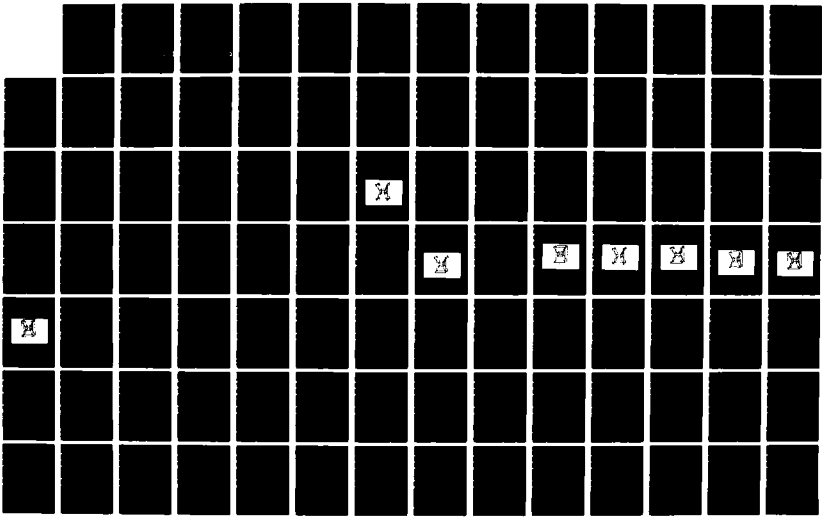
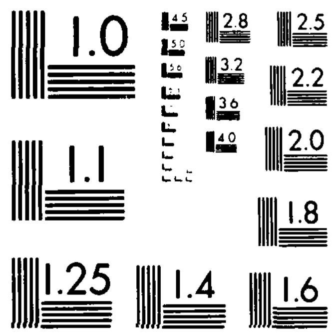
AD-A189 573 A SIMULATION STUDY OF A PARALLEL PROCESSOR WITH UNBALANCED LOADS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING 1/2

UNCLASSIFIED T S MOORE DEC 87 AFIT/GCS/ENG/87D-20 F/G 12/6 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DTIC FILE COPY

AD-A189 573

DTIC
ELECTE
MAR 0 2 1988
S H

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

88 3 1 189

AFIT/GCS/ENG/87D-20

A SIMULATION STUDY OF A PARALLEL PROCESSOR WITH UNBALANCED LOADS

THESIS

Timothy Scott Moore
Captain, USAF

AFIT/GCS/ENG/87D-20

AFIT/GCS/ENG/87D-20

A SIMULATION STUDY OF A PARALLEL PROCESSOR WITH UNBALANCED LOADS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Information Systems

Timothy Scott Moore, B.S.
Captain, USAF

December 1987

Acknowledgments

I have had the good fortune to have a thesis topic that I really enjoyed working on. Consequently, many of the long hours spent working on this product did not seem like work. It is doubtful that any thesis is a one person effort, and this one is certainly no exception. I owe a debt of gratitude to a number of people. First and foremost, is my thesis advisor, Captain Wade Shaw, USA, who treated me more like a research partner than a student working for him. Thank you, Sir.

Secondly, I thank Captain Cathy Lamanna, USA, for her help with benchmarking the hypercube to determine message transmission times. I thank First Lieutenant Steve Wagner for his help with LaTex and for still being around at 3:00 a.m. I thank LTC Robinson, LTC Valusek, and Major Litko of the Operations Research Department for their help and comments pertaining to the animated simulation. I thank Major Woffinden for his comments as one of my committee members.

Last, but certainly not least, I thank my beautiful wife Julie for all her support and understanding during our 18 months at the Air Force Institute of Technology.

Timothy Scott Moore

Table of Contents

List of Figures

List of Tables

Abstract

The purpose of this study was twofold; first, to estimate the impact of unbalanced computational loads on a parallel processing architecture via Monte Carlo simulation and second, to investigate the impact of representing the dynamics of the parallel processing problem via animated simulation. The study is constrained to the hypercube architecture in which each node is connected in a predetermined topology and allowed to communicate to other nodes through calls to the operating system. The routing of messages through the network is fixed and specified within the operating system. Message transmission preempts nodal processing causing internodal communications to complicate the concurrent operation of the network.

This study defines two independent variables, the degree of imbalance and the degree of locality. The degree of imbalance characterizes the nature or severity of the load imbalance and the degree of locality characterizes the node loadings with respect to node locations across the cube. A SLAM II simulation model of a generic 16 node hypercube was constructed in which each node processes a predetermined number of computational tasks, and, following each task, sends a message to a single randomly chosen receiver node. An experiment was designed in which the independent variables, degree of imbalance and degree of locality were varied across two computation-to-IO ratios to determine their separate and interactive affects on the dependent variable, job speedup.

ANOVA and regression techniques were used to estimate the relationship between load imbalance, locality, the computation-to-IO ratio, and their interactions to job speedup. The results show that load imbalance severely impacts a parallel processor's performance. The effect of locality is minor and enters the speedup model primarily as an interactive term; suggesting that the locality effect on speedup is dependent on the degree of imbalance. The intensity of IO is significant and affects speedup across all levels of locality and imbalance.

An animated simulation was developed using The Extended Simulation System (TESS) and the SLAM II model mentioned previously. The animation was designed such that a 16 node hypercube structure was displayed. The processing nodes and channels were displayed in different colors to represent specific types of processing. Watching the animation execute proved useful in two ways. First, the animation was useful in visually explaining the concepts of imbalance and locality. Secondly, and most importantly, the animation was valuable as a means of verifying the underlying simulation model.

A SIMULATION STUDY OF A PARALLEL PROCESSOR WITH UNBALANCED LOADS

1. Introduction

1.1 Background

The advent of multiprocessor computer systems has resulted in evidence of decreased processing time for jobs that can be decomposed into parallel processes. This phenomenon has been tested to reveal significant but not perfect increases in process speedup as additional processors are added. This is particularly true for loosely-coupled systems in which inter-node communications overhead does not allow an N node parallel processor to achieve the theoretical linear speedup. That is, an N node machine actually produces something less than an N times speedup. Speedup is defined as the ratio of the single processor execution time to the time measured with additional processors.

Multiprocessing computing systems are divided into two general categories, tightly-coupled systems and loosely-coupled systems. Tightly-coupled systems usually have a large, shared memory through which the individual processors communicate. In loosely-coupled systems, each processor has its own local memory. An individual processor and memory module form a processing element, and the processing elements are connected through an interconnection network. The processors communicate with each other via messages sent through the interconnection network. An emerging, loosely-coupled architecture showing promise is the hypercube machine discussed by Wiley (1). A 16 node hypercube is depicted in Figure 1.

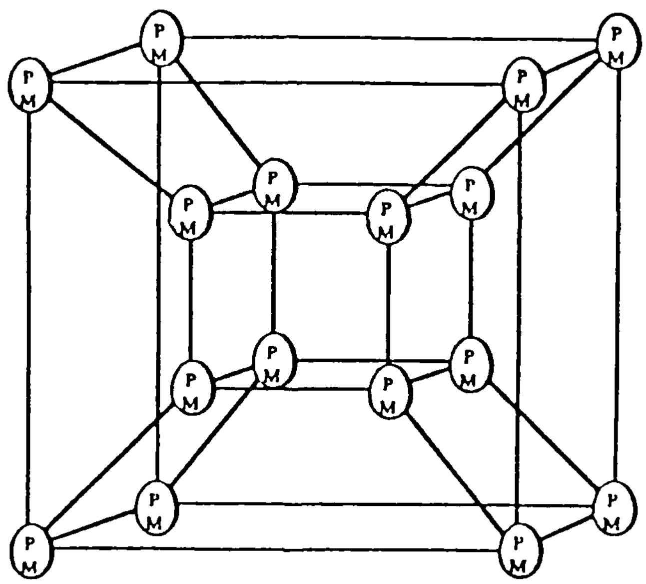


Figure 1. 16 Node Hypercube

1.2 Problem Statement

Programming for a parallel system requires that the programs be decomposed into parallel processes. It is intuitive that decomposing a program such that the processing nodes are evenly balanced in terms of the workload will produce the optimum results. However, it will not always be possible to achieve perfect node balancing. Therefore, a specific concern of parallel system users is the effects of processor load balance, and the distribution of the balance (spatial locality of the load), on the performance of a job. This concern is important because the effects of load balancing will significantly affect the choice of decomposition algorithms.

The purpose of this thesis is to determine the effect of processor load balance on the speedup of a process executed in parallel on a loosely-coupled multiprocessor computer system.

1.3 Scope

This thesis specifies a means of characterizing processor load balance and spatial locality. Monte Carlo simulation is used to determine the effect of the load balance on the speedup of a job executed in parallel on a loosely-coupled system. Since the relative impact of communication time between nodes is known to dramatically affect performance, the experiment is conducted at two levels of CPU/IO intensity to insure that the effects of imbalance are isolated.

The investigation is limited to the performance of a 16 node hypercube machine with statistically controlled processor and IO loads. This approach does not necessarily predict the performance of any particular algorithm. Rather, it is intended to develop a fundamental relationship between processor load balance, load locality, and speedup. This relation provides insight that explains the general nature of workload partitions and locality.

In addition to the discrete event simulation experiment, the effectiveness of animated simulation is investigated using The Extended Simulation System (TESS). This thesis does not consider how to choose a decomposition algorithm; only the effects of choosing a poor decomposition algorithm.

1.4 Approach

This thesis investigates the effectiveness of simulation and animation to illuminate the relation of non-homogeneous processor loads to job execution time. The steps followed are given below.

1. Determination of a topology and message routing algorithm: A 4-D, 16 node Intel iPSC Hypercube topology is used in which each node is connected to four other nodes according to the Gray code. Message routing between nodes is fixed in accordance with the Intel Hypercube iPSC operating system.
2. Determination and characterization of the independent variables:
   (a) The primary independent variable is the degree of processor load imbalance. The degree of imbalance is characterized by the coefficient of variation of the individual processor loads. This metric is computed as

   $$B = \sigma_b / \mu_b \tag{1}$$

   where σ_b is the standard deviation of the processor loads and μ_b is the mean. The greater the variation in loads, the greater the degree of imbalance. For a perfectly balanced system, the degree of imbalance is zero.

   (b) A secondary independent variable is locality. The concept of locality is used to characterize the node loadings with respect to node location. For example, assume that nodes 0 and 1 each have 45% of the load of a given job, and the remaining 10% is distributed evenly among the other nodes. This loading scheme will be characterized by a value for the degree of

imbalance and a value for the degree of locality. Now, assume the same case except that nodes 0 and 15 each have 45% of the load. In this case, the degree of imbalance will be the same, but the degree of locality will be different because nodes 0 and 15 are not directly connected as is the case for nodes 0 and 1. Locality is characterized by calculating L_i for each node and calculating the coefficient of variation of the L_i's. L_i is calculated by the equation

$$L_i = \sum_{j=0}^{15}(l_{i,j} * p_j), \forall i, i \neq j \tag{2}$$

where $l_{i,j}$ is the number of hops required to transfer a message from node i to node j and p_j is the percentage of the total load computed by node j. If a message is sent from a node to an adjacent node, then the transmission requires one hop. If the originating and receiving node are separated by one intermediate node, then the transmission requires two hops.

3. Construction of the simulation model: A model of a 16 node hypercube was constructed using the SLAM II (2) simulation language. Each node executes a predetermined number of cpu bursts, where following each cpu burst, a message is sent to a randomly determined recipient node. I/O packet sizes are uniformly distributed between 100 and 1,024 bytes. Processor bursts are exponentially distributed with a mean of R times the average message transmission time, where R is the predetermined CPU/IO ratio and set at two values of 2 and 10.

4. Determination of message transmission times: The model constructed requires an equation for message transmission times based on bytes of data transferred. The message transmission time equation was determined by running a benchmark program on the Intel Hypercube and performing regression analysis. These results are presented in Chapter 3.

5. Design of the experiment: An experiment was constructed in which the degrees of imbalance and locality were varied across two levels of CPU/IO processing

ratios (R) in order to determine their independent and interactive effects on job speedup.

6. Investigation of the effectiveness of animated simulation: The model was animated using The Extended Simulation System (TESS) in order to determine if the real-time graphical output of an animated simulation provides additional insight into a complex problem that may not be discerned from the textual output generated by the discrete event simulation.

7. Characterization and presentation of results: The relationships between the degree of imbalance, locality, and job performance are characterized and presented by testing the research hypotheses which contrast system performance with controlled, experimental factors. The research hypotheses are stated as follows:

   - HO1: There is no variability in process speedup explained by the degree of load imbalance, the locality of the load imbalance, the processing to communication ratio, or any interaction on a hypercube parallel processing machine.

   - HO2: Animated simulation does not provide additional insight into a complex problem that cannot be discerned from the textual output of a discrete event simulation.

1.5 Overview

Chapter Two presents a summary of current research in the field of multiprocessor computers. Chapter Three presents the methodology, the development of the model, the design of the experiment, and the development of the animation. Chapter Four presents the results of the discrete event simulation experiments and the animation. Chapter Five summarizes the thesis and presents the final results.

2. Literature Review

There are many factors against which multiprocessor performance can be evaluated. Recent performance evaluations have studied the effects of workload mix, program behavior, processor interconnection networks, redundant interconnection networks, memory management, and decomposition strategies. However, all of these studies were performed with balanced processor loads.

Nestle and Inselberg (3) have shown that a tightly-coupled multiprocessor system can be modularly expanded while providing strictly linear improvements in performance. These improvements, they claim, are independent of the workload mix. They contrast their results to loosely-coupled multiprocessor systems which, they claim, cannot sustain linear increases in performance when running nonhomogeneous workloads due to the interprocessor communication overhead. (3:233) The claim about the performance of loosely-coupled systems is indirectly related to the research goal of this thesis and is of considerable interest and importance. Although the claim is intuitive, no study was cited to su port their claim.

Du (4) performed a study where system structure and program behavior were the two main factors. This study, Du claims, is set apart from others by the fact that previous studies have usually ignored program behavior. His study evaluated the performance of a multiprocessor in which a crossbar was employed to interconnect p processors to m commonly shared memory modules. A set of nonuniformly distributed probabilities, including a probability, P(0), which represents a processor not generating any request, was used to model the program behavior. However, no distinction was made between processors. Several relations between the average processor utilization, average request completion time, and the effective memory bandwidth were obtained. Processer utilization, P_u, is defined as $P_u = b + L_o$ where b is the memory bandwidth and L_o is the average number of processors which do

not generate any nonlocal requests. The relations developed are given below:

$$b = p * (1 - P(0))/P(0) + (1 - P(0)) * T), \tag{3}$$

$$L_o = b * P(0)/(1 - P(0)), \tag{4}$$

$$T = (p/b - (P(0)/(1 - P(0))), \tag{5}$$

where T is the average request completion time of a nonlocal request and p is the number of independent processors. (4:462)

Bhuyan (5) evaluated two loosely-coupled architectures, each having three types of interconnection networks: shared bus, crossbar, and a class of multistage interconnection networks called Omega networks. The probability that a message is accepted was used as a measure of the performance. The study showed that for a high rate of internal requests, an Omega network performed close to a crossbar, but at a considerably reduced interconnection cost. (5:256)

Padmanabahn and Lawrie (6) conducted an evaluation which focused on the effect of redundant path interconnection networks on performance. Their evaluation showed that redundant path networks provide significant fault tolerance at a minimal cost. In addition, improvements in performance and very graceful degradation were shown to result from the availability of redundant paths. (6:117)

Jalby and Meier (7) conducted a study in which memory management was the primary factor. They claim that as the memory organizations of large multiprocessor computers become more complex, data management in the memories becomes a crucial factor for achieving high performance. An architecture which combines vector and parallel capabilities on a two-level shared memory structure was studied via analyzing and optimizing matrix multiplication algorithms. The optimized algorithms yielded high efficiency kernels which can be used for many numerical algorithms such as LU and Cholesky factorizations. (7:429)

Gerhinger, Segal, Siework, and Vrsalovic (8,9) present a model for predicting multiprocessor performance on iterative algorithms based on the decomposition

strategy used. Each iteration was assumed to require some amount of access to global data and some amount of local processing. The application cycles were allowed to be synchronous or asynchronous, and the processor may or may not have incurred waiting time, depending on the relationship between the access time and the processing time. The amount of global data accessed and the processing time incurred by the parallel processes were dependent upon characteristics of the algorithm and its decomposition. The decomposition of several algorithms was studied and several decomposition groups were identified. The Poisson partial differential algorithm was used to determine how decomposition affected the performance of the algorithm. (8:396) This study is more directly related to the research topic than the others presented. However, the decompositions that were evaluated resulted in balanced loads on the individual processors and the system evaluated was a tightly-coupled system.

Wiley (1) claims that an evenly distributed load is essential for efficient parallel computing. In addition, factors such as communication time between processors are also important. While these claims are intuitive, no references are cited to support the statements.

Reed and Grunwald (10) performed an evaluation on the Intel iPSC which relates directly to this thesis effort. They determined the message processing times for nearest neighbor nodes on the iPSC Hypercube. They characterized the transmission times in accordance with the following model:

$$S = L + Nt \tag{6}$$

where S is the transmission time, L is the communication startup time (latency), t is the transmission time per byte, and N is the number of bytes transferred. They performed a least-squares fit of the data to the linear model with the following results:

$$L = .0017 seconds$$
$$t = .00000283 seconds$$

This evaluation is duplicated in this thesis and the results are compared.

As the research cited indicates, there are many factors against which multiprocessor performance can be evaluated. One such factor is the effect of processor load balance on performance. The effect of the load balance will be important in determining which algorithm to use when decomposing programs into parallel processes. It is accepted that perfect balancing results in more efficient program execution. However, the effects of imbalanced processor loads has not been thoroughly researched and characterized. Consequently, there is minimal literature pertaining directly to the subject. There is, however, a considerable amount of literature which evaluates the effects of other factors on performance. These factors include workload mix, program behavior, processor interconnection networks, redundant processor interconnection networks, memory management, and decomposition strategies. These factors represent the state- of-the-art in multiprocessor performance evaluation.

It is intuitive to suspect that a parallel processor will exhibit reduced speedup as the degree of load imbalance is increased to the extent that the execution time resembles the performance of a smaller machine. The major issue is the nature and the severity of the load imbalance and locality effect; and, whether that effect is consistent across different processing to communication ratios.

3. Research Method

The purpose of this thesis is to determine the effects, if any, of processor load imbalance, locality, and their interaction on speedup. In order to investigate the effects of load balance, it is necessary to develop load balance and locality metrics. These definitions were provided in Chapter 1. Using these metrics, an experiment design was set up so that the metrics were varied over a sufficiently wide range to observe the impact on process speedup. Since the metrics are quantitative, regression techniques were used to determine the nature and significance of the main and interactive terms.

3.1 Model Construction

A simulation model was developed using the SLAM II simulation language. The model simulates generalized processing on a 4-D, 16 node Hypercube in which each node executes a predetermined number of processor bursts. Following each burst, a message is sent to one random receiver node. Single receivers were chosen over multiple receivers so that IO processing would not dominate the execution time. Random (uniform) receivers were chosen so that communications would be evenly distributed across the entire cube. Additionally, it was not within the scope of this research to model processor affinity with regard to IO.

3.1.1 Message Transmission Times. A crucial aspect of this research was to model the time required to transmit a message between nodes. In the case of nearest neighbor transmissions this problem has been researched as shown in Equation 6. However, this thesis must simulate transmissions between non-nearest neighbors as well as nearest neighbor nodes. Since Equation 6 was estimated based on nearest neighbor transmissions, and does not account for any intermediate processing time at nodes along the sender/receiver path, it cannot be used for the purpose of this

study. Therefore, an equation for message transmission had to be estimated which accounted for intermediate node processing.

The simulation model treats message transmission as a series of one or more direct node communications. The initial sending node performs some amount of I/O overhead (S) and transmits the message. The time required to transmit the message is dependent on the size, in bytes, of the message (X). If the receiving node is the final destination, then some amount of final receiving I/O overhead (R) is performed and the message is terminated. If the receiving node is not the final destination, then some amount of intermediate node I/O overhead (I) is performed, the next receiver node is determined, and the message is transmitted to that node.

Based on this model of message sending, the total time required to send a message (T) can be expressed as

$$T_{ms} = \beta_0 + (\beta_1 HX) + (\beta_2 I) + error \tag{7}$$

where β_0 is the sum of S and R, H is the number of hops between the initial sender and the final receiver, X is the number of bytes in the message, β_1 is the overhead per byte of data transferred, I is the number of intermediate nodes visited, and β_2 is the overhead associated with each intermediate node.

In order to determine actual message passing times, a benchmark program was executed on the Intel iPSC Hypercube. Node 0 sent and received a message of fixed length, ranging from 5 to 1024 bytes, to and from nodes 1 thru 31. The program is constructed so only one message is being passed at a time. For each unique receiver node, 20 data points were collected. Each data point is the average of the time required for node 0 to send and receive a message 100 times (200 total transmissions) to and from the receiver node. The output data set consisted of 620 times, 20 for each receiver node. Included with each time was the number of intermediate nodes passed through to the receiver node.

Equation 7 was estimated using linear regression. The data set, SAS (11) program, and regression results are given in Appendix A. A plot of the data is shown in Figure 2.

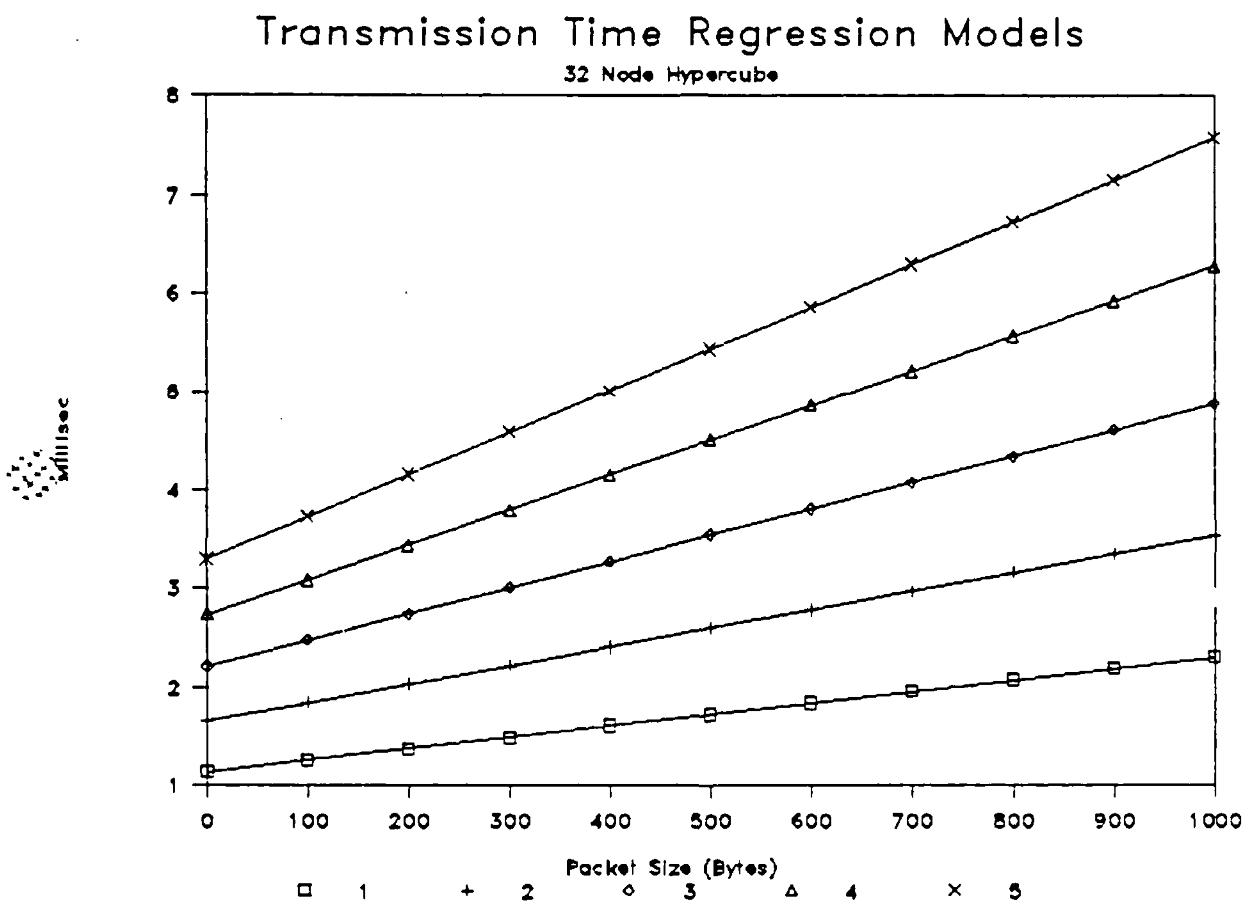


Figure 2. Plot of Message Transmission Data

The estimation of Equation 7 yielded the following relation:

$$T_{ms} = 1.23 + 0.000897HX + 0.485I \tag{8}$$

The model's adjusted R-Square was 0.9939 and each coefficient significant at the 99% level. The latency of 1.23 ms is lower than the 1.7 reported by Reed and Grunwald (10) and the 0.897 microseconds per byte is considerably higher than their estimate. These differences are attributed to the fact that Reed and Grunwald confined their estimation to nearest neighbor transmissions only, as well as possible enhancements to the Hypercube since their study. The 0.485 millisecond delay experienced at each intermediate node represents the low level protocol to hand-off the message to another communications channel and is not dependent on message length. This time is somewhat lower than the latency time at the sender and receiver ends of the path but represents a major culprit in explaining the less than theoretical speedup obtained in practice.

3.1.2 Model Design. Using Equation 8 as the function which maps the message length to transmission time, the simulation model described below was constructed. The hypercube is modeled as a single user system with 16 nodes declared in the cube. The cube and the 16 nodes are unique SLAM Resources while communication channels are modeled as single server Activities preceded by a Queue. Each channel uses a unique activity number and queue file number which facilitates routing of entities through the network via a lookup table. Basically, the simulation proceeds as follows:

1. A job enters the system and waits for the cube.
2. When the cube becomes available, it is allocated to the first waiting job.
3. The time the cube is allocated is recorded as the job start time.
4. The job is replicated into 16 processes.

5. Each process is assigned a processor identification, a number of processor bursts, and a process burst duration.
6. Each process waits for the node to which it is assigned.
7. When the node becomes available, the node processes one burst of exponentially distributed length and initiates a single I/O of random length (100-1024 bytes). The number of bursts remaining for that node is decremented by one.
8. The node that processed and initiated the I/O is freed.
9. The process entity is replicated to become a message entity. The process entity returns to wait for the node to become available so it can execute another burst.
10. A random receiver node ID is assigned to the message entity.
11. A table look-up is used to determine the channels and intermediate nodes required to send the message to its destination node.
12. The message waits in the appropriate channel QUEUE.
13. When the channel service activity becomes available, the message is transmitted. The message transmission time is dependent upon the number of bytes of data transferred in the message.
14. The receiving node is preempted.
15. If the receiving node is not the final destination node, it processes the message as an intermediate node, determines the next node and channel, and retransmits the message. The intermediate node is freed.
16. If the receiving node is the final destination, it processes the message as the destination node, the destination node is freed, and the message entity is terminated.
17. When all bursts have been completed and all messages have been processed, the time the job has been in the system is collected and the cube is freed.

The flow diagram of the simulation model described above is given in Figures 3 and 4. The SLAM II code for the model is given in Appendix B.

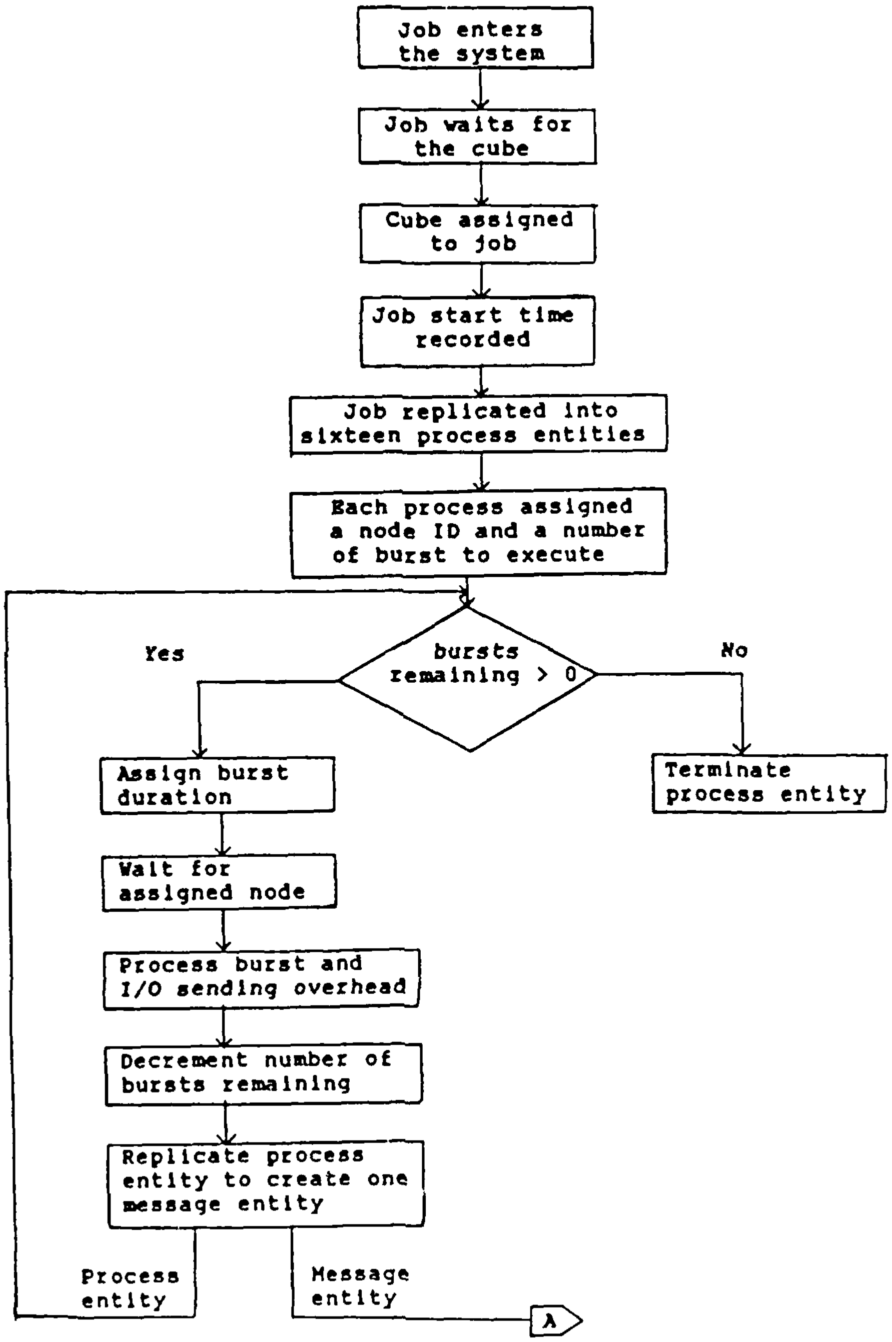


Figure 3. Simulation Model Flow Diagram (a)

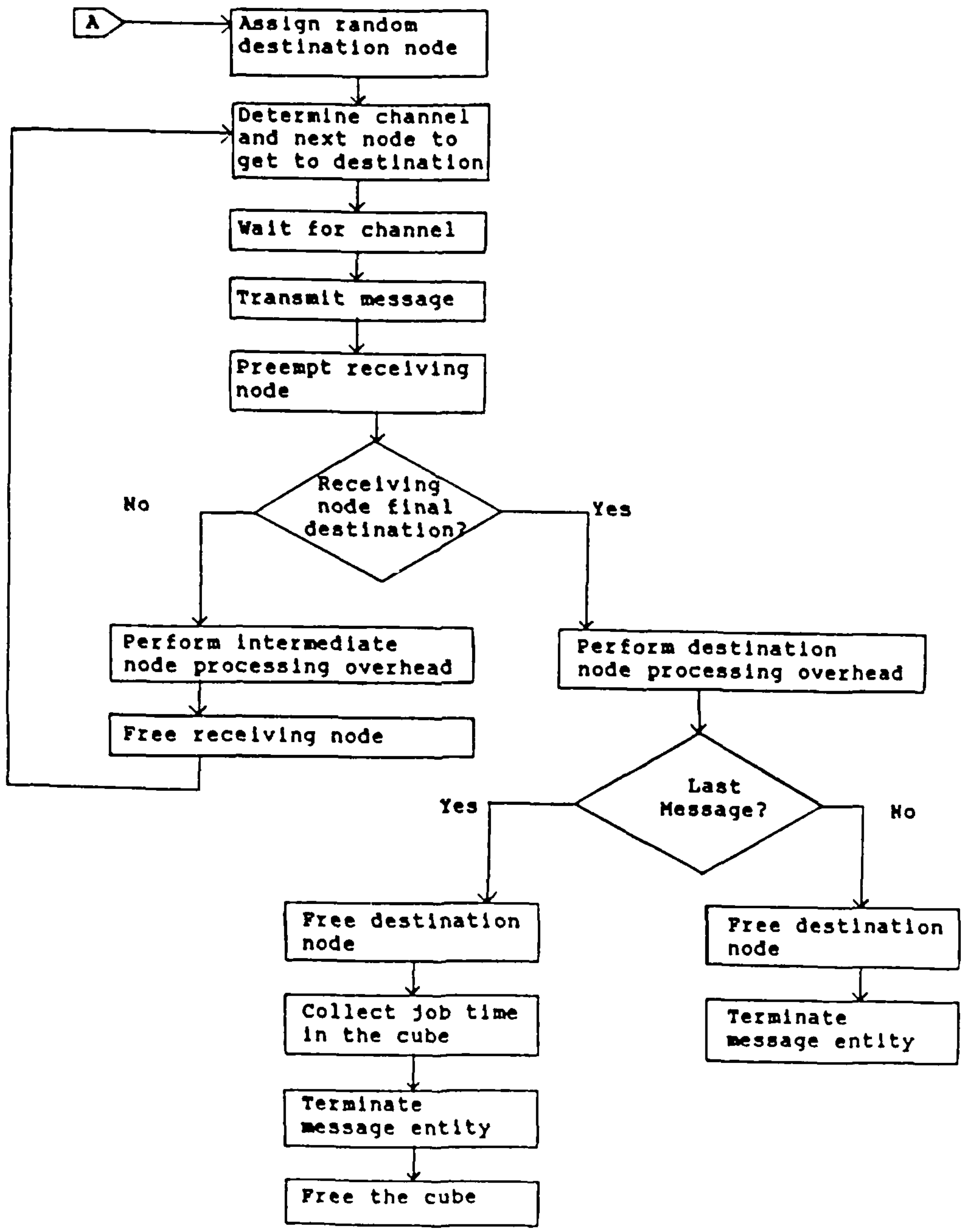


Figure 4. Simulation Model Flow Diagram (b)

3.2 Experiment Design

An experiment was designed and used to reduce experimental error. The imbalance and locality metrics were varied across two levels of CPU/IO ratios. The general linear model of the experiment is:

$$S = \mu + R + B + L + RB + RL + BL + RBL + error \tag{9}$$

where S represents the observed process speedup, μ is the experiment average, R is the ratio of average processor burst time to average message transmission time, B is the load imbalance metric, L is the locality metric, and RB, RL, BL, and RBL are the interactions of these terms.

3.3 Analysis of Data

Table 1 shows the experimental data. Each test case was simulated for R values of 2 and 10. Data was obtained by setting the total number of processor bursts for a generic process to 256 where each burst was distributed as a negative exponential with a mean of 3.23 milliseconds for R=2 and 16.14 milliseconds for R=10. The IO time was set to the random variable determined by the length of a message distributed uniformly between 100 and 1024 bytes and the timing equation given in Equation 8. The degrees of imbalance and locality corresponding to the cases given in Table 1 are given in Table 2.

Table 1. Experiment Design Node Loadings

Node Number

Case	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
2	24	22	20	18	16	14	12	10	1	11	13	15	17	19	21	23
3	1	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
4	32	30	28	26	1	3	5	7	17	8	6	4	2	27	29	31
5	1	2	3	4	5	6	7	8	32	31	30	29	28	27	26	17
6	38	6	6	6	6	38	6	6	38	6	6	38	6	6	6	38
7	38	38	38	6	38	6	6	6	38	6	6	6	6	6	6	6
8	72	8	8	8	8	8	8	8	8	8	8	8	8	8	8	72
9	72	72	8	8	8	8	8	8	8	8	8	8	8	8	8	8
10	79	7	7	7	7	7	7	7	7	7	7	7	7	7	7	79
11	79	79	7	7	7	7	7	7	7	7	7	7	7	7	7	7
12	100	4	4	4	4	4	4	4	4	4	4	4	4	4	4	100
13	100	100	4	4	4	4	4	4	4	4	4	4	4	4	4	4
14	121	1	1	1	1	1	1	1	1	1	1	1	1	1	1	121
15	121	121	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	241	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 2. Test Case Degree of Imbalance(B) and Locality (L)

Case	B	L	Case	B	L
1	0.000	0.000	9	1.366	0.224
2	0.368	0.021	10	1.537	0.000
3	0.368	0.088	11	1.537	0.252
4	0.776	0.052	12	2.049	0.000
5	0.776	0.186	13	2.049	0.335
6	0.957	0.065	14	2.562	0.000
7	0.957	0.194	15	2.562	0.419
8	1.366	0.000	16	3.750	0.484

3.4 Data Collection

Each experimental unit (composed of a degree of imbalance, degree of locality, and burst to message time ratio) was simulated so that batch means of 10 runs with 10 jobs each were used to obtain an execution time average. In all. 3200 jobs were simulated. It is noteworthy that an additional case exists which is not shown in Table 1 which represents the single processor case where only one node is loaded with all the processor bursts. This case corresponds to a single processor machine with a known behavior of $256x3.23 = 826.4$ millisecond execution time for R=2 and 4132 milliseconds for R=10. Case 1 represents the perfectly balanced case where B and L are 0.

3.5 Validation

The resulting simulated job execution times were considered to be accurate reflections of actual hypercube performance for several reasons. First, the balanced case measurements were reasonable and correspond to actual experience with the hypercube. Second, when the degree of imbalance was maximized the execution time did in fact move towards the known uniprocessor time. Third, the progression of execution times as the load imbalance was increased was reasonable and produced a speedup profile which agrees with engineering judgement and intuition. Finally, each component of the simulation was tested and desk checked to insure compliance with the design specifications.

3.6 Animation

The discrete event simulation experiment provided some interesting results which are presented in the following chapter. In order to answer the second research hypothesis, pertaining to the effectiveness of animated simulation, the SLAM model of the generic 16 node hypercube (described in Figures 3 and 4) was animated using The Extended Simulation System (TESS). TESS is a graphics based interactive

system installed on the Classroom Support Computer (CSC), which is a VAX 11-785 running under the VMS Version 4.5 operating system.

Because animated simulation is a relatively recent development, and its usefulness is a function of the user's ability to analyze the animation as he watches it execute, the evaluation of this technology was rather subjective in nature.

3.6.1 Animating With TESS. TESS allows the user to either graphically build a SLAM II network using the Network Builder or link an existing SLAM II source file. Since the simulation model had already been constructed for the discrete event simulation experiment, the TESS Network Builder was not used.

TESS provides concurrent animation and post-simulation animation capabilities. In the concurrent animation mode, the model is animated as the simulation executes. In the post-simulation mode, a history file is built as the simulation executes and the animation is executed later from the history file. A history file may be created from a concurrent animation which allows for subsequent post- simulation animations. For the purposes of this thesis, post-simulation animation was used. A post-simulation animation requires the specification of a facility, a set of rules, and a history file.

3.6.1.1 History File. Special TESS commands must be inserted into the SLAM II network code to collect information for the animation and history file. The commands required for the animation used in this thesis are presented and described in Appendix C. An example of a history file is also given in Appendix C.

3.6.1.2 Facility. The facility, built using the TESS Facility Builder, is the background on which the animation executes. The facility built and used for this thesis is shown in Figure 5.

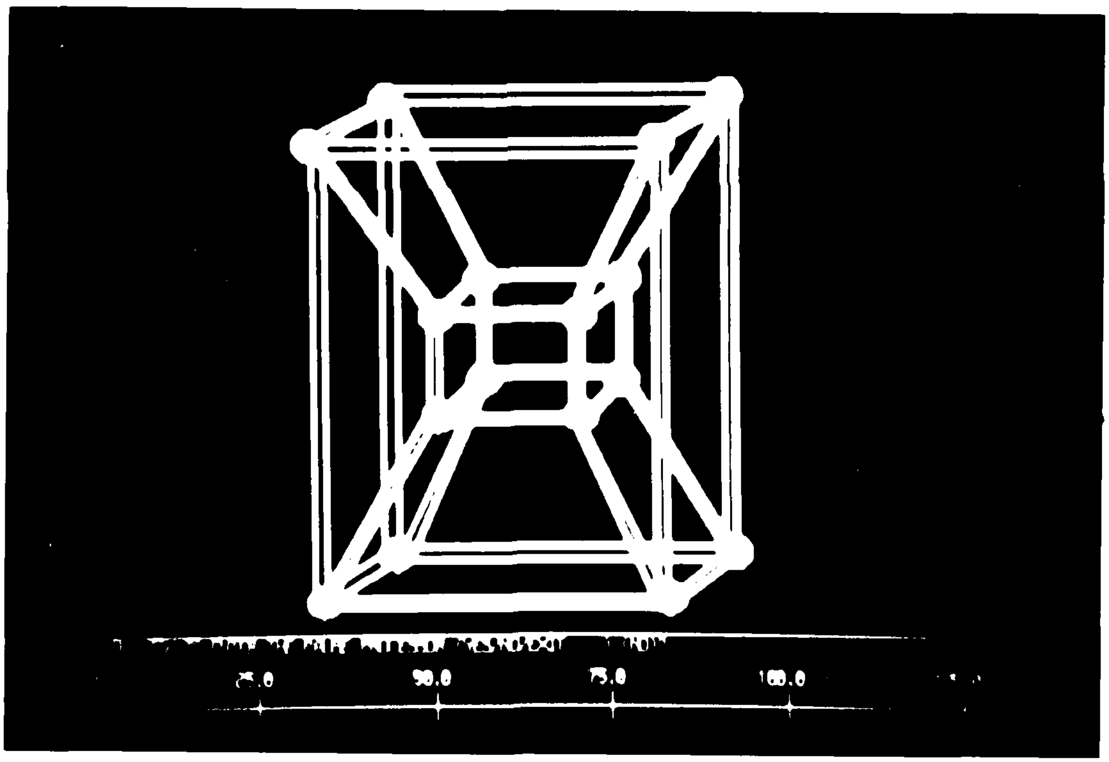


Figure 5. Hypercube Facility

Each node is represented as a circle icon with a unique name. The channels connecting adjacent nodes are represented as path icons. The Intel iPSC Hypercube has full duplex channels connecting adjacent nodes. The full duplex allows a node to simultaneously receive data from and transmit data to the same adjacent node. A single full duplex channel is modeled as two half duplex (uni-directional) channels. For example, the single full duplex channel connecting nodes 1 and 2 is represented as two half duplex channels; one connecting node 1 to 2, and the other connecting node 2 to 1. Each channel icon has a unique named based on the nodes it connects. The names of the two channels mentioned above are "C1X2" and "C2X1" respectively.

3.6.1.3 Rules. The rules are built with the TESS Rule Builder and govern the display of the animation. The rules specify how to display the facility, and when and how to color or move specific icons. The rule set used for this thesis is given in Appendix D. The facility is initially displayed with all icons (nodes and channels) colored white indicating they are idle. When a processor node executes a processor burst the node is colored red. When a processor node is performing message processing the node is colored green. When the node is idle it is colored white. When a communications channel is busy it is colored blue; otherwise, the channel is colored white.

3.6.2 Animation Experiment. The goal of this portion of the thesis is to determine if an animated simulation provides additional insight that could not be discerned from the discrete event simulation. Unfortunately, the usefulness of the animation is a function of the user's ability to evaluate the executing animation. Therefore, this portion of the thesis required a rather subjective approach of "watch the animation and see what it tells us".

Three of the test cases developed for the discrete event simulation experiment (refer to Table 1) were animated. They were cases 1, 8, and 9. Case 1 represents the perfectly balanced system. Cases 8 and 9 were chosen because they represent a medium degree of imbalance (B = 1.37) at two degrees of locality (0.00 and 0.22, respectively).

4. Results

This chapter presents the results of the discrete event simulation experiment and the animations. Statistical analysis is used to evaluate the experimental model given in Chapter 3 with respect to the data generated by the discrete event simulation. The model is refined to remove nonsignificant terms and the resulting models are presented. The effects of load imbalance (B), locality (L), and the processor to communication ratio (R), and any significant interactions are discussed with respect to the models.

4.1 Discrete Event Simulation Results

The raw execution times and the speedup statistics are shown in Table 3. Figure 6 depicts speedup with respect to B, the load imbalance metric. It is evident that *extreme variability is present and that there is overwhelming evidence of nonlinear effects*.

Figure 6 indicates that a 16 processor hypercube with a degree of imbalance of 1.5 and a CPU/IO ratio of 10 performs like the theoretical 4 processor machine. Clearly, the penalty for load imbalance is severe.

Table 3. Discrete Event Simulation Results

	Input		Time (ms)		Speedup	
Case	B	L	R=10	R=2	R=10	R=2
1	0.00	0.00	429.2	110.2	9.6	7.5
2	0.37	0.02	503.2	127.8	8.2	6.5
3	0.37	0.09	496.3	129.4	8.3	6.4
4	0.78	0.05	652.5	164.4	6.3	5.0
5	0.78	0.19	664.0	167.9	6.2	4.9
6	0.96	0.06	779.8	188.7	5.3	4.4
7	0.96	0.19	780.9	194.8	5.3	4.2
8	1.37	0.00	1306.0	303.7	3.2	2.7
9	1.37	0.22	1309.0	322.5	3.2	2.6
10	1.54	0.00	1417.0	331.5	2.9	2.5
11	1.54	0.25	1442.0	350.2	2.9	2.4
12	2.05	0.00	1792.0	413.1	2.3	2.0
13	2.05	0.34	1818.0	438.8	2.3	1.9
14	2.56	0.00	2162.0	493.5	1.9	1.7
15	2.56	0.42	2178.0	525.8	1.9	1.6
16	3.75	0.48	4075.0	936.6	1.1	0.9
Uni	4.00	0.52	4131.8	826.4	1.0	1.0

B = Degree of Imbalance
L = Degree of Locality
R = Ratio of Computation Processing to Message Processing

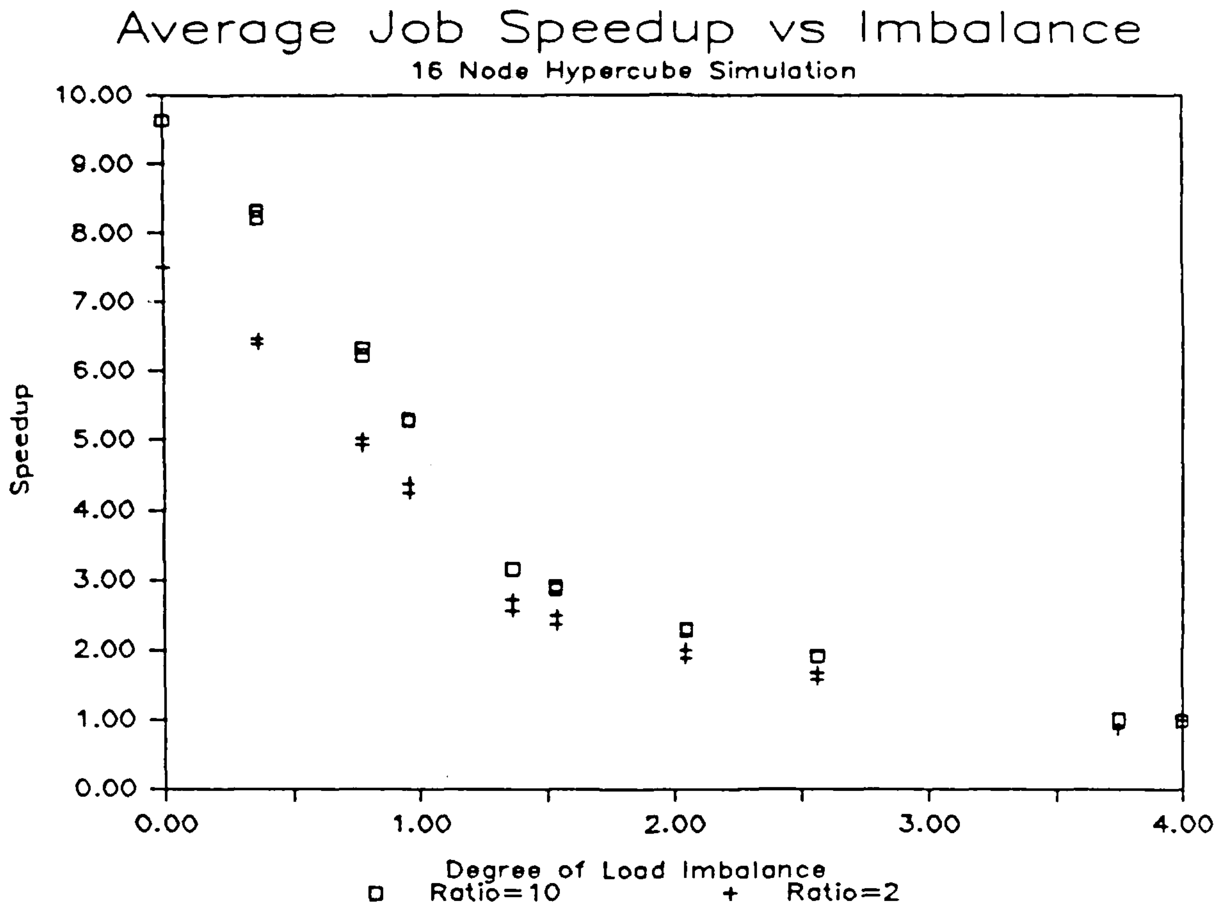


Figure 6. Average Speedup Measurements versus Degree of Imbalance (B)

Figure 7 displays speedup with respect to the locality metric (L). Again, the effect is evident and nonlinear.

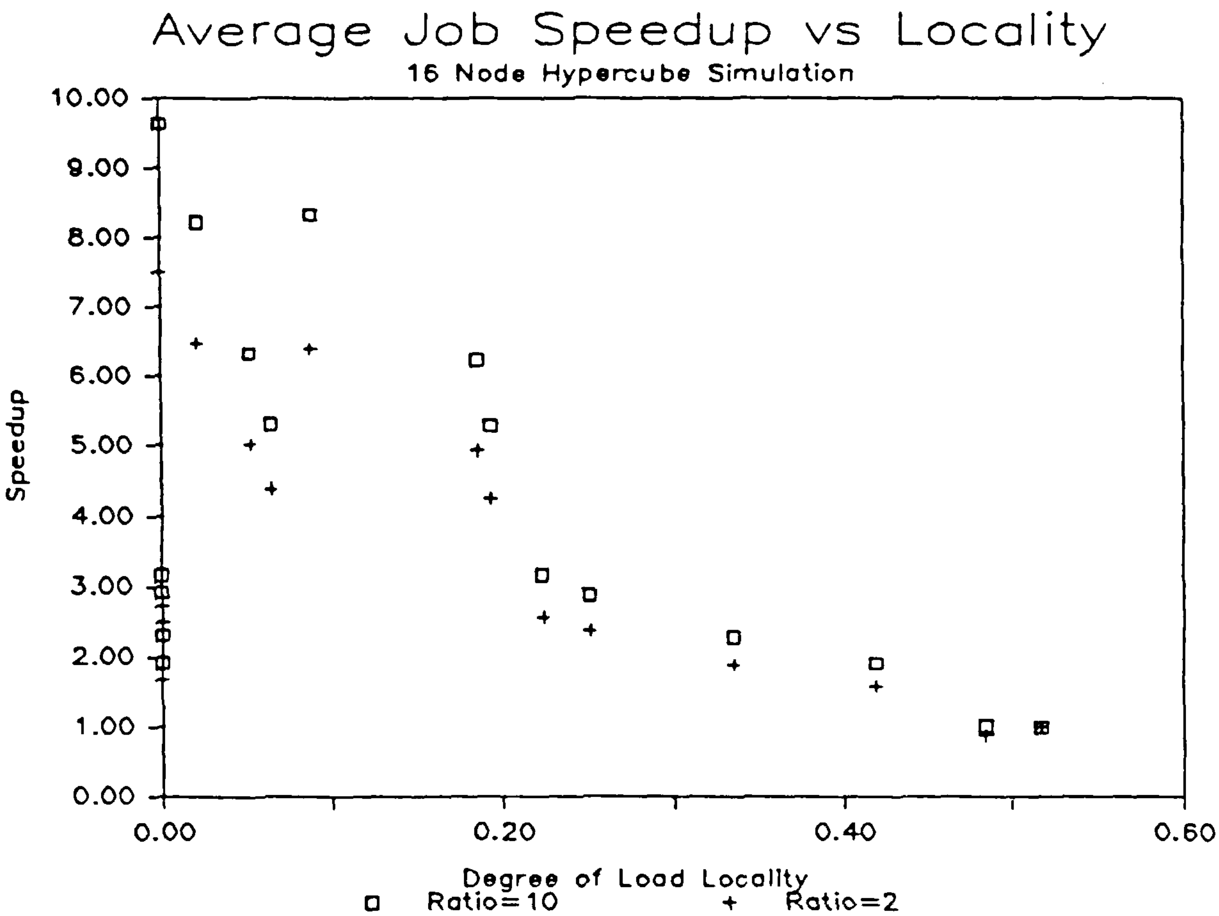


Figure 7. Average Speedup Measurements versus Degree of Locality (L)

Due to the nonlinear nature of Figures 6 and 7, the inclusion of a nonlinear term in Equation 9 was necessary. Therefore, the square of B and L were introduced into the model as a simple way to estimate nonlinear effects and restate the hypothesized relationship to be a polynomial fit of degree 2. The resulting relationship is given below.

$$\begin{aligned} S &= \mu + R + B + B^2 + L + L^2 + RB + \\ & \quad RL + BL + RBL + error \end{aligned} \tag{10}$$

Other transformations could have been used; however, the curvature of the lines appear to obey a power law which is straightforward in its estimation.

Equation 10 was estimated using least squares. This is refered to as Model 1. Each term from Equation 10 that was not significant at the 99% level was removed. The resulting relation, refered to as Model 2, was re-estimated. Again, the nonsignifcant terms of Model 2 were removed and the resulting relation, Model 3, was re-estimated. Table 4 shows the estimated coefficients.

Table 4. Model Coefficients
Least Square Estimates

Term	Model 1	Model 2	Model 3
Constant	**7.46**	**8.16**	**8.13**
R	**0.25**	**0.10**	**0.10**
B	**-4.89**	**-4.91**	**-4.84**
L	*1.54*		
RXB	*-0.11*		
RXL	-0.25		
BXL	**-5.93**	-0.30	
RXBXL	0.15		
B^2	**1.05**	**0.80**	**0.74**
L^2	*29.57*		
Model R	0.984	0.961	0.960

Italic Significant at 0.05 level
Bold Significant at 0.01 level

Figure 8 compares the observed data with predictions based on Model 1, the full featured model. The model predicts speedup quite well as evidenced by an R-square value of 98.4%.

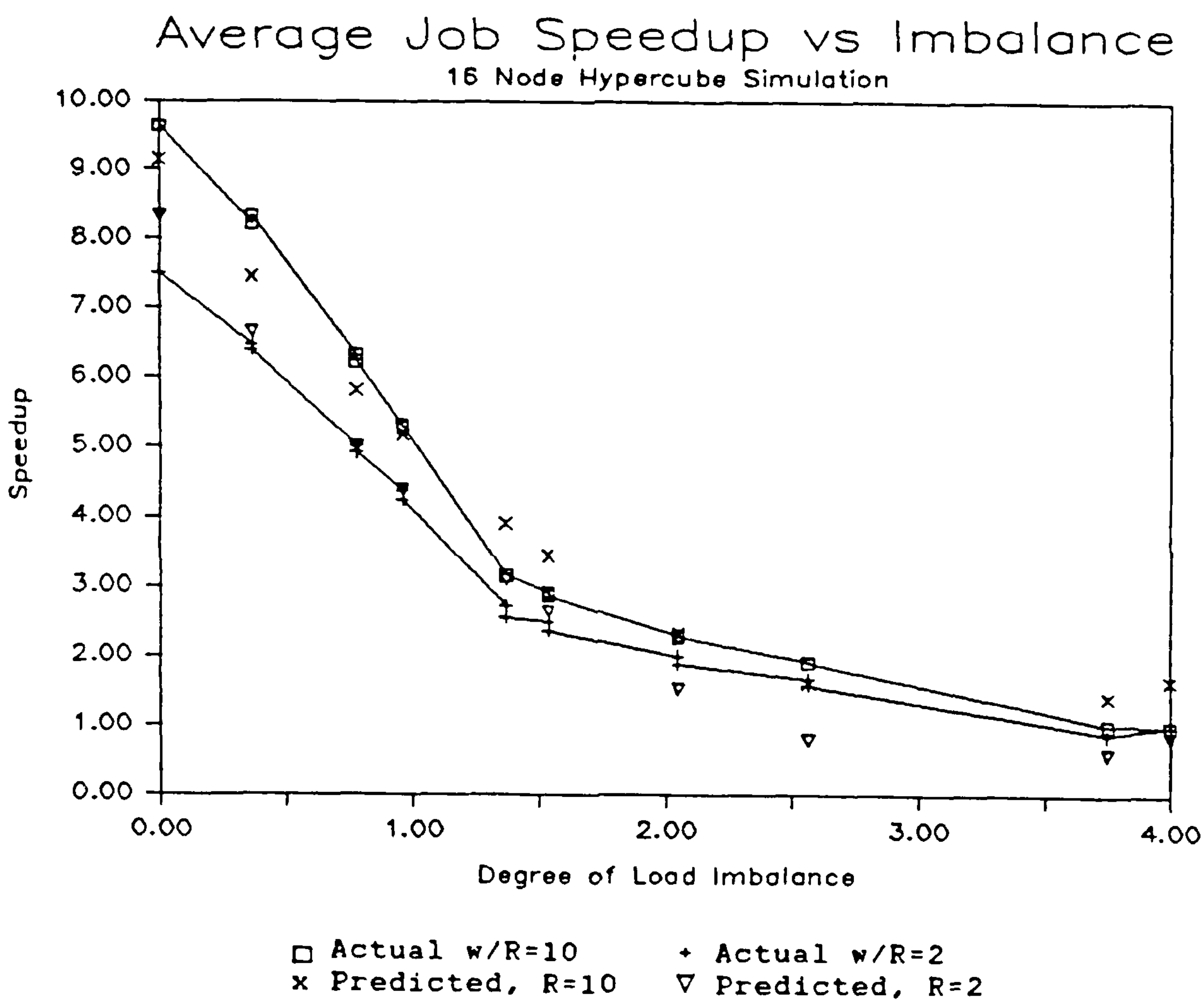


Figure 8. Actual Speedup versus Model 1 Predictions

Considering terms significant at the 95% level. Model 1 establishes that the ratio of processor burst time to message processing time is highly significant but not really involved in any interaction. That is, the ratio's effect is a scaler which tends to adjust the curve up or down by a factor of .25 milliseconds per unit of R. The balance and locality metrics both enter the model as linear and nonlinear operators. The impact of locality appears to be minimal and involved in a balance interaction. Apparently, locality alone does not influence speedup to any great extent. The impact of locality was investigated by varying the locality over four settings at two settings of imbalance (B=1.37 and B=2.05) for both values of R. Figure 9 indicates confirmation of the regression analysis: locality does not affect speedup very much!

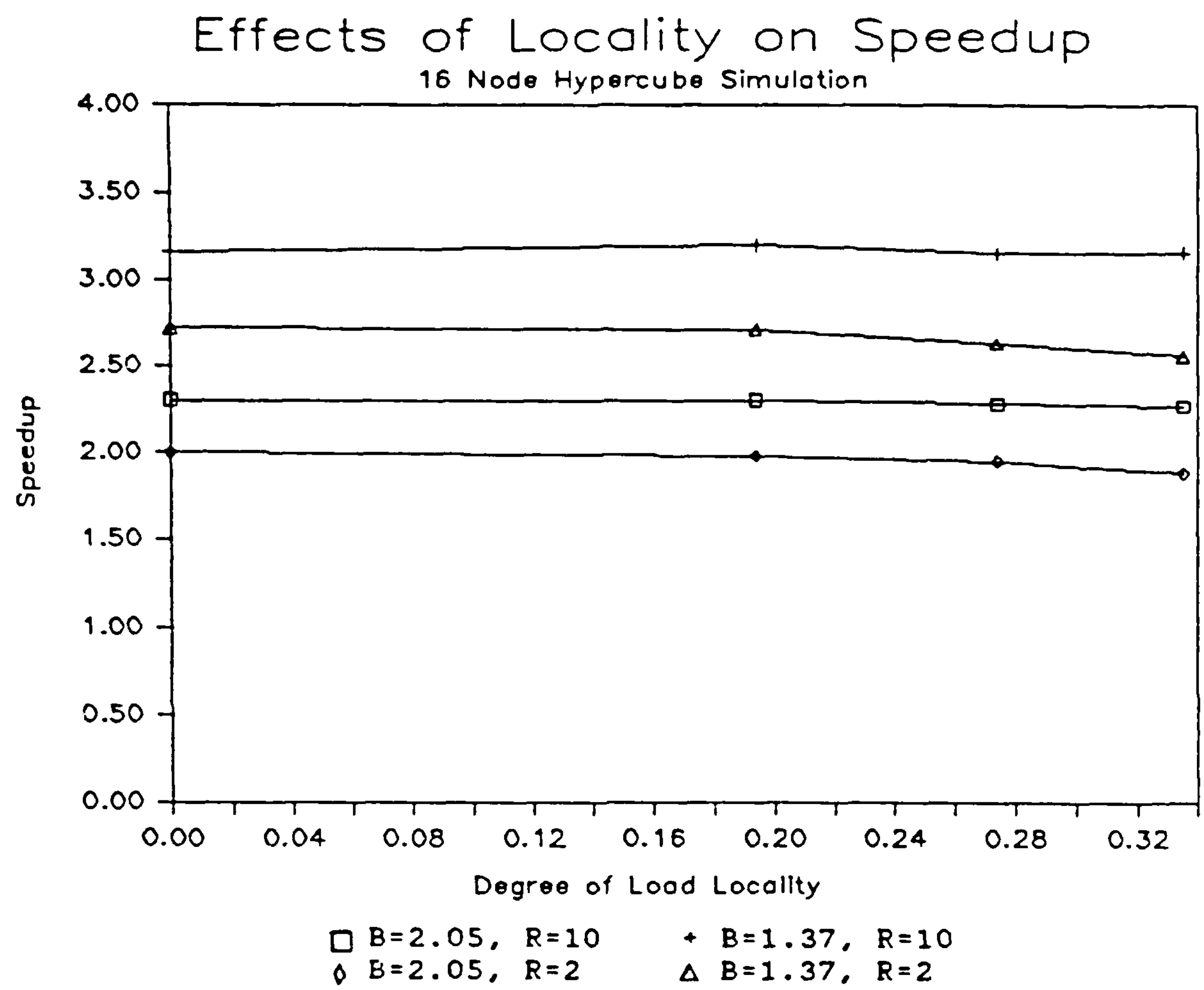


Figure 9. Effects of Load Locality

Using the results of Model 1, the nonsignificant terms were removed to yield the simpler Models 2 and 3. The Models' R-square remained high (96.1 and 96.7) indicating little loss of explanatory power as terms are removed. Figure 10 depicts the actual observations versus predictions using the simplest model, Model 3.

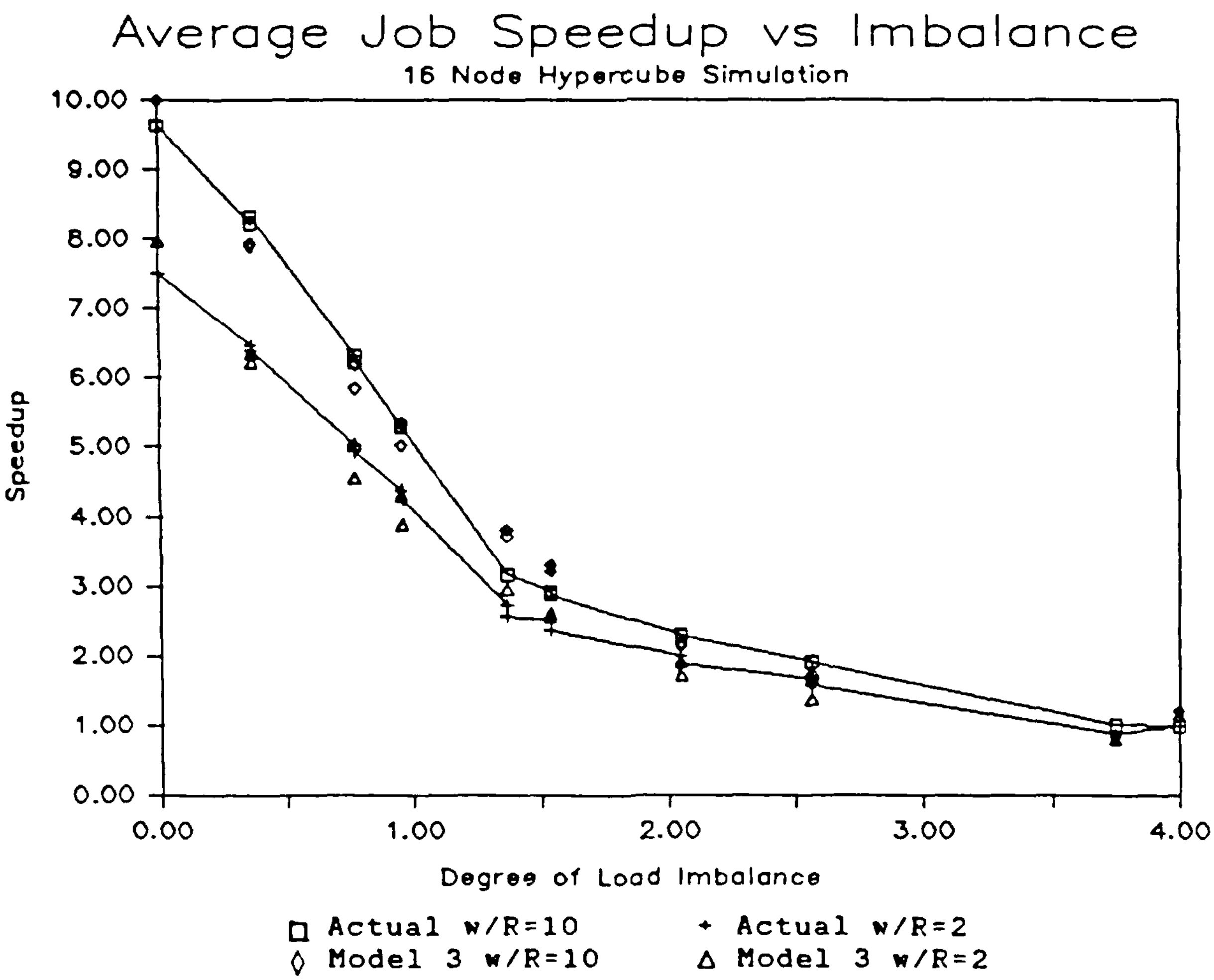


Figure 10. Actual Speedup versus Model 3 Predictions

Model 3 is given as:

$$S = 8.13 + 0.10R - 4.84B + 0.74B^2 \tag{11}$$

Interpretation of Model 3 is straightforward: the ratio of processor time to message time contributes 0.1 milliseconds per unit across all levels of imbalance; the imbalance metric (B) basically governs the shape of the speedup degradation by subtracting out 4.84 milliseconds for every unit increase in B adjusted by adding the square of B times 0.74 milliseconds. The penalty for imbalance is severe initially but tapers off as the square term adds back the speedup as the degree of imbalance increases. For example, in the case of R=2, increasing B from 0.0 to 0.5 results in a reduction of speedup of 1.5 (7.5 to 6.0). However, increasing B from 2.0 to 2.5 results in a reduction of speedup of only approximately 0.25 (2.00 to 1.75).

4.1.1 Evaluation of First Research Hypothesis. Recalling that the imbalance metric is the ratio of the load standard deviation to the load average; it appears that as the standard deviation reaches the hypercube average (B=1), performance suffers dramatically. Furthermore, as the IO load becomes more dominant (lower R value), the speedup is initially worse and subject to the same imbalance phenomenon. Locality appears to be of minimal impact and involved in statistically significant interactions which are difficult to explain from an engineering point of view. In short, the first research hypothesis (HO1) is soundly rejected. There is definitely a relationship between load balance, locality, and the IO intensity which characterizes speedup phenomenon very well.

4.2 Animation Results

As discussed in Chapter 3, evaluating the results of an animated simulation is not necessarily straightforward. The usefulness of the animation depends upon the viewer's ability to evaluate the animation as it executes. This ability is, in turn, dependent on the viewer's knowledge of the problem domain, the system being simulated, and the simulation model itself.

For this thesis, the difficulty of evaluating the animation is further compounded by the fact that the CSC is a multi-user, time-shared system. Ideally, an animation should run from start to finish, with no interruptions. This uninterrupted processing should allow the viewer to develop a time frame reference with regard to the animation. A realistic time frame reference enables the viewer to accurately determine how long certain aspects of the animation take compared to others; which aides in developing a realistic understanding of the entire system being animated.

Unfortunately, on a multi-user, time-shared system, the TESS user must compete for CPU time with the other system users. Consequently, the animation executes for intermittent CPU time slices, during which times the animation is updated. After a CPU time slice, the animation remains static until the next allotted time slice. The time between CPU time slices is dependent upon the load on the system. The result is an animation which, in terms of real clock time, takes longer to execute as the system load increases. This dependency on system load makes it difficult, if not impossible, to develop a reasonable time frame reference for the animation. This inability to establish a time frame reference makes it difficult to compare the animations of different system loadings (test cases).

A final problem relates to the presentation of the results. That is, how does one present the results of an animated simulation within the text of a thesis? This problem is approached using two different methods. First, pictures of the three test cases animated are presented. Second, the summarized opinions of faculty and students who viewed the animations are presented.

4.2.1 Pictorial Representation. The test cases animated were cases 1, 8, and 9. Each animation case represents the simulated execution of only a single job with the node loadings given in Table 1 for the particular case. Consequently, the stated turnaround times for the cases may differ from those given in Tabel 3 which represent the averaged results from 100 jobs.

Case 1 represents the perfectly balance case, B=0 and L=0. The job required 105.2 ms of simulation time to complete. Figure 11 shows the state of the animation at approximately 50 ms into the animation.

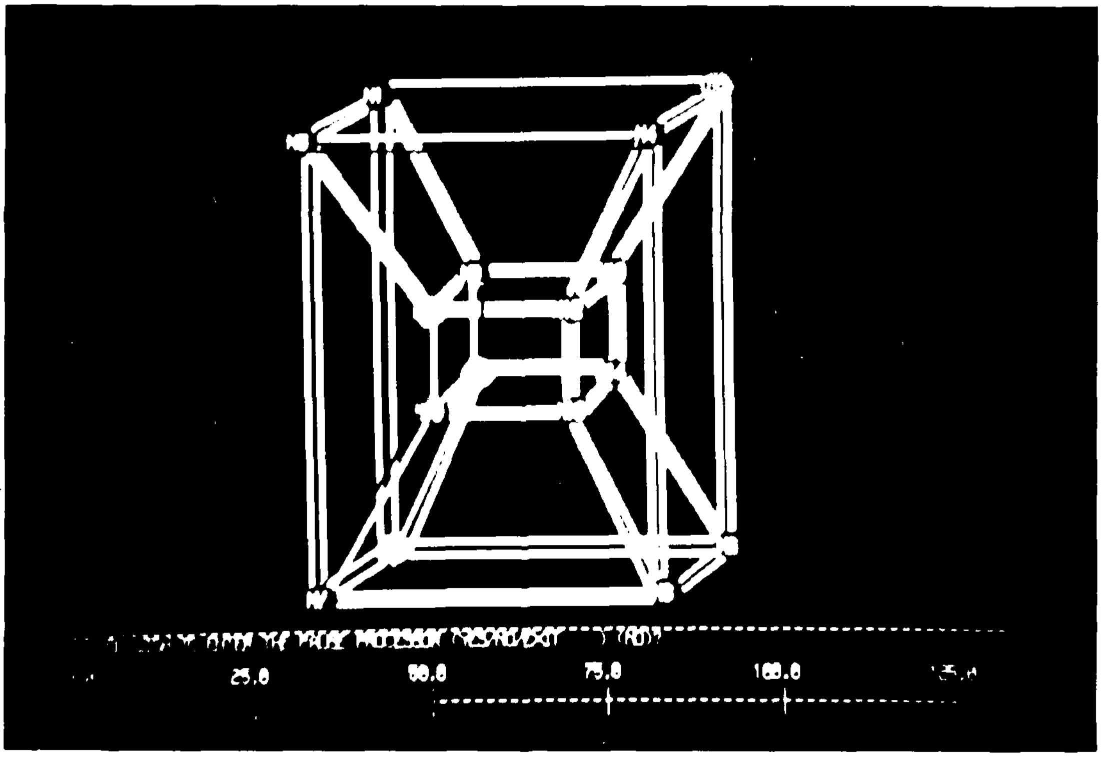


Figure 11. Case 1: 50 ms into Animation

The time-line displayed at the bottom of the screen is automatically generated by TESS. At this point in the animation, all 16 nodes are busy performing either computational processing or message processing, and 5 channels are busy transmitting data. Nodes 1, 3, 4, 6, 7, 8, 9, 10, 12, 14, 15, and 16 are red, indicating they are performing computational processing. Nodes 2, 5, 11, and 13 are performing message processing, indicated by green. The blue channels connecting the node pairs 1 and 2, 3 and 4, 7 and 15, 9 and 13, and 11 and 15 indicate that data is being passed from one of the nodes to the other. The case 1 animation remains balanced in processing, all nodes remain busy, until approximately 75 ms into the animation, by which time, some of the nodes have completed their assigned number of bursts and remain idle except for message processing.

Case 8 represents a degree of imbalance of 1.37 and a degree of locality of 0.00. The job required 272.3 ms of simulation time. Figure 12 shows the state of the animation at approximately the start of the animation.

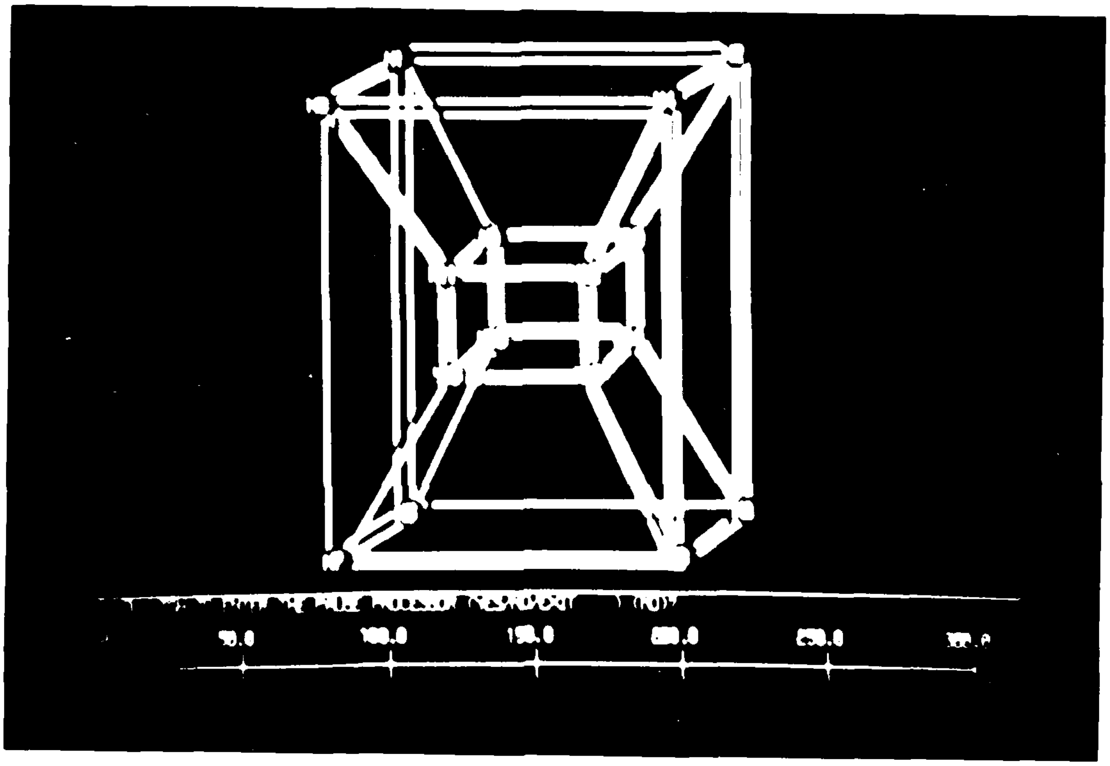


Figure 12. Case 8: Start of Animation

Notice the change in the time-line. At this time all 16 nodes are performing computational processing and 5 channels are transmitting data. Figure 13 depicts the same animation at approximately 35 ms into the simulation.

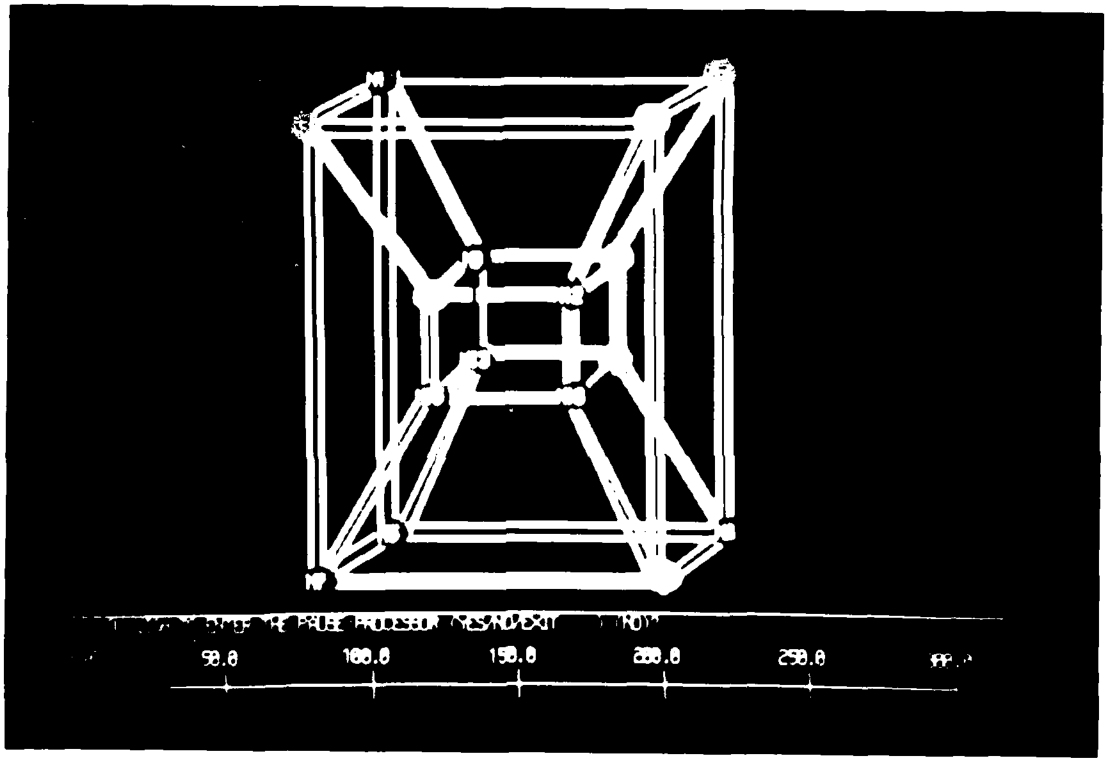


Figure 13. Case 8: 35 ms into Animation

Nodes 4, 8, and 11 have completed their assigned number of processes and are idle. The remaining nodes are performing either computational or message processing. Figure 14 shows the status of the animation approximately 50 ms into the simulation.

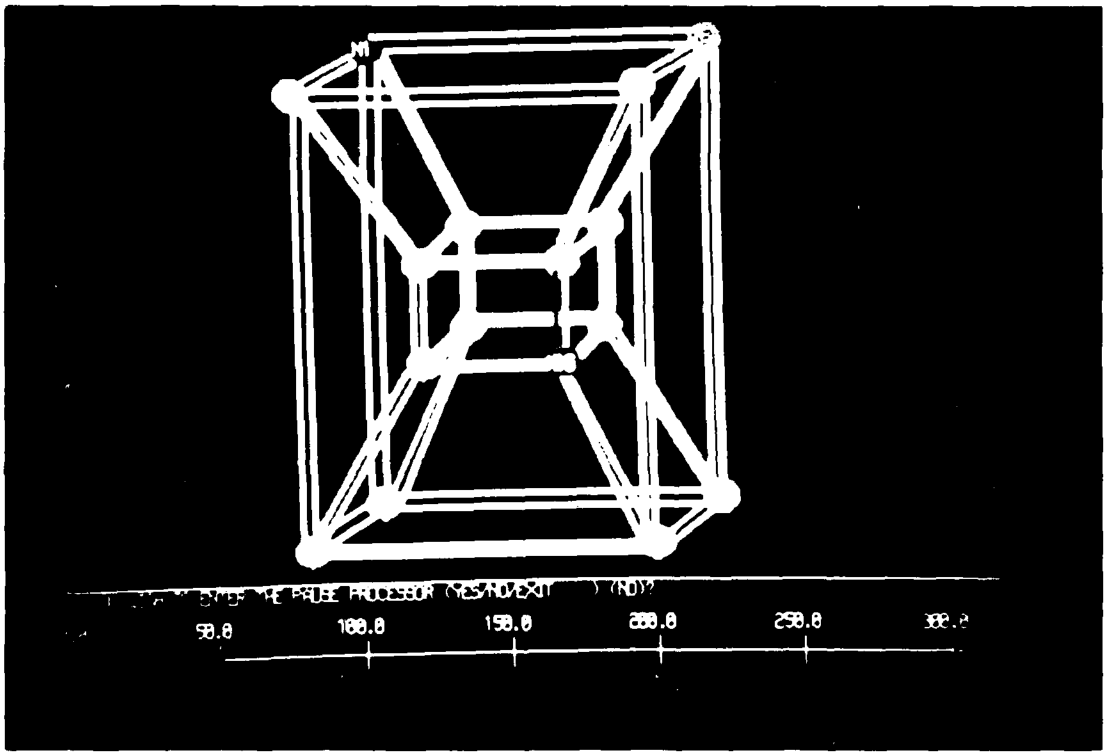


Figure 14. Case 8: 50 ms into Animation

By 50 ms into the simulation, all nodes except 1 and 16 have completed their assigned number of processes. Nodes 1 and 16 are still performing computational processing, nodes 2 and 14 are performing message processing, and node 16 is transmitting a message to node 12.

Case 9 represents a degree of imbalance of 1.37 (same as case 8) and a degree of locality of 0.22. This job requires 312.6 ms of simulation time to complete. Figures 15, 16, and 17 show the animation at approximately the start, 100 ms, and 135 ms into the simulation, respectively.

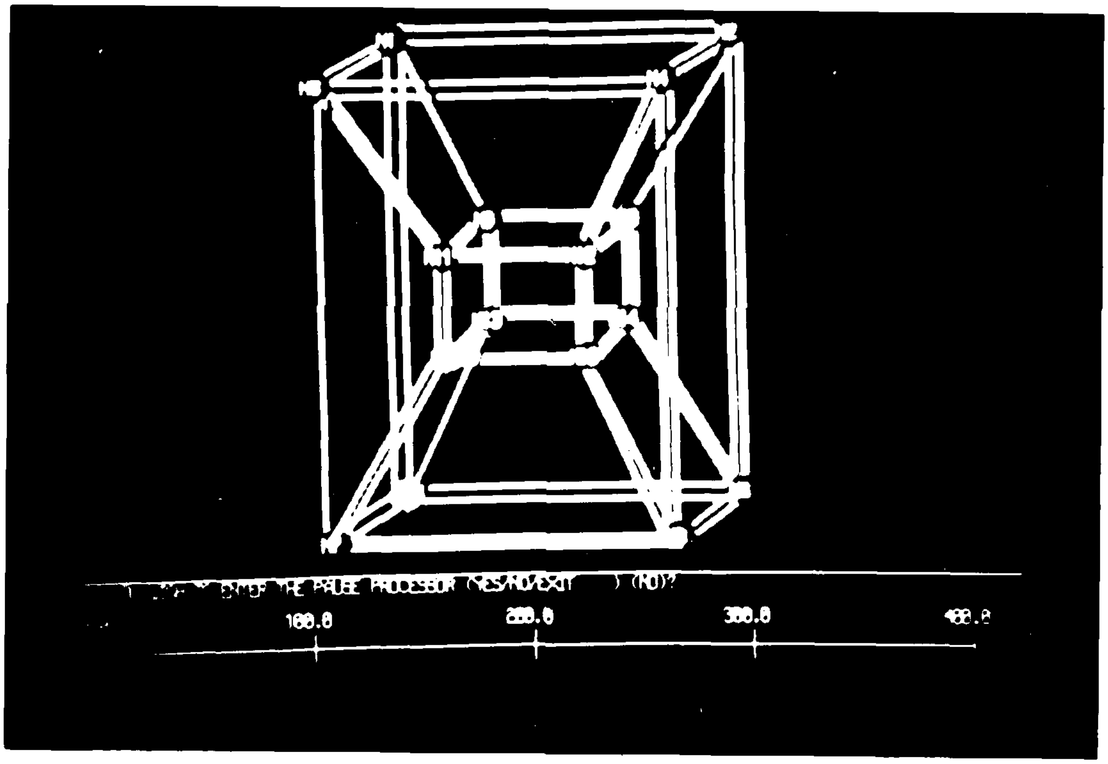


Figure 15. Case 9: Start of Animation

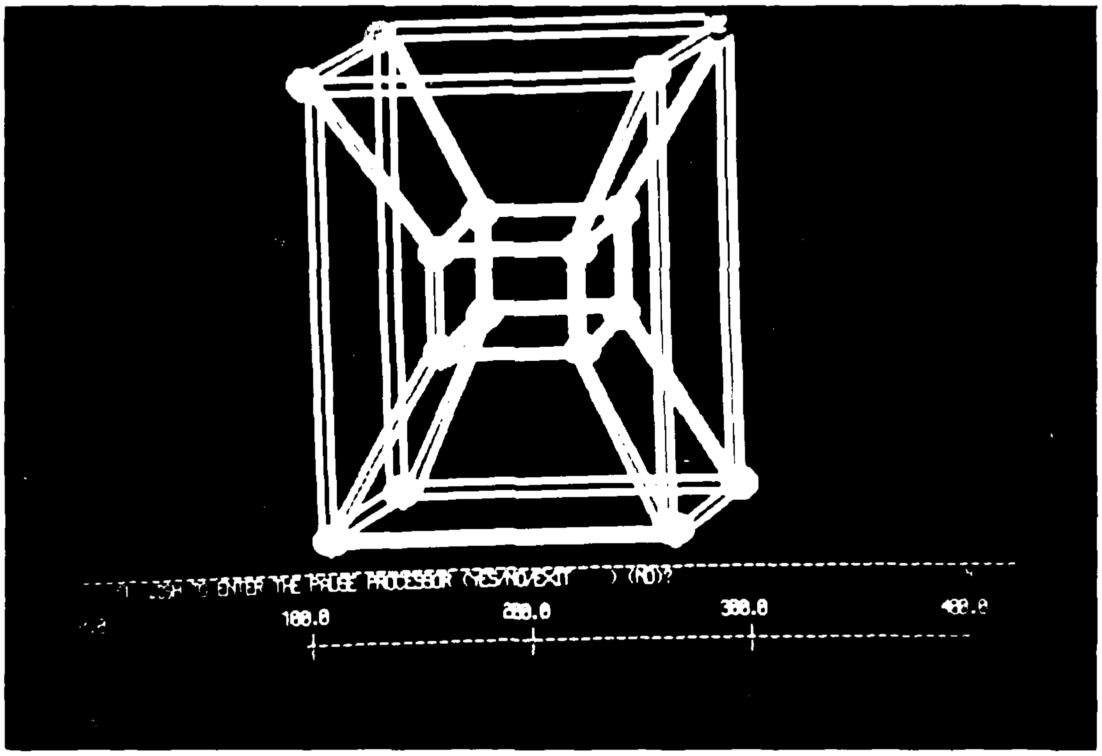


Figure 16. Case 9: 100 ms into Animation

Figure 17 is particularly interesting. At 135 ms into the animation, all nodes except 1 and 2 have completed their assigned number of processes and are idle. However, nodes 1 and 2 have each pre-empted each other from performing computational processing because they each sent a message to the other. It is increased pre-emption, due to the proximity of the heavily loaded nodes, that causes case 9 to take longer than case 8.

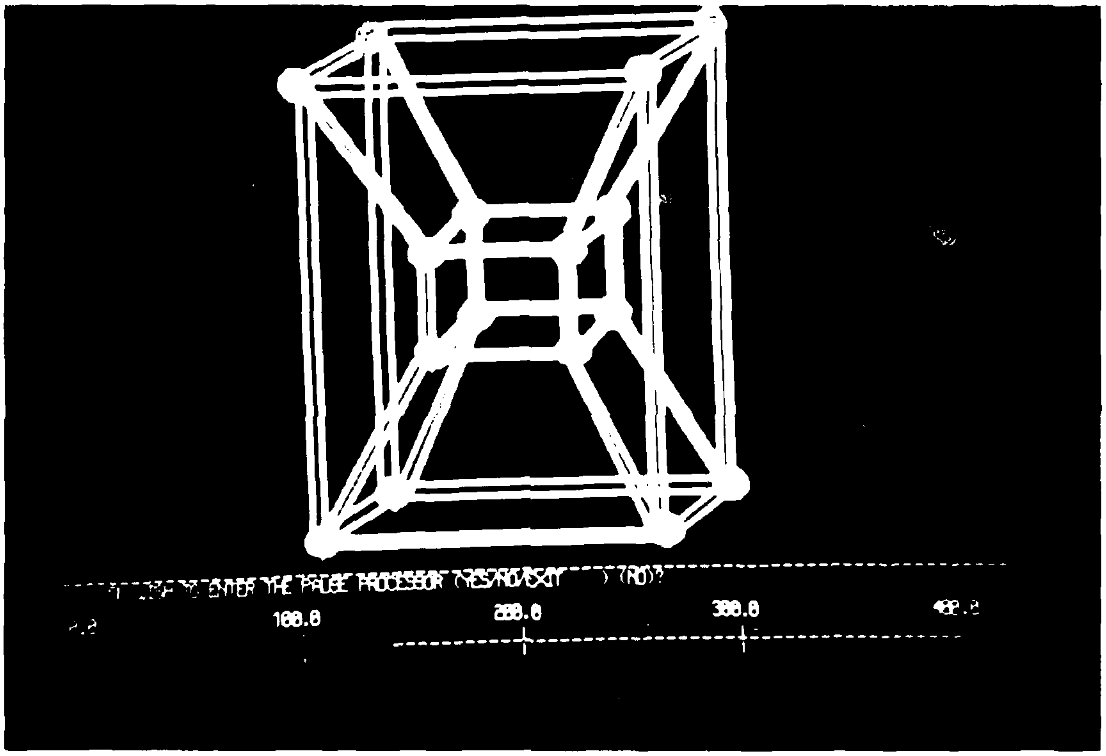


Figure 17. Case 9: 135 ms into Animation

4.2.2 Viewer Evaluation. The animated simulations were viewed and evaluated by faculty and students in the Operations Research and Electrical and Computer Engineering departments. These two departments were chosen for the following reason. Members of the Operations Research department, while having little or no knowledge of the operation of a hypercube, are very familiar with simulation and the capabilities of TESS. Conversely, members of the Electrical and Computer Engineering department have a well developed understanding of the hypercube, but are generally not familiar with TESS. Selection based on this reasoning provided for evaluation from two fundamentally different perspectives.

The animation was explained to each viewer and the viewer was asked his impressions of the animation. Each viewer was asked if they thought the animation was useful for the particular problem being studied; that is, to determine the effects of load imbalance and locality on speedup. The remarks for each department were summarized and are presented in the following paragraphs. A discussion of these opinions is given following both summarizations.

4.2.2.1 Operations Research. The animations are useful for introducing the hypercube architecture to the uninitiated viewer. The animations were particularly useful in visually explaining the concepts of the imbalance and locality metrics, and allowing the viewer to graphically see the impact of the internodal communications.

However, the animations did have their drawbacks. When viewers tried to watch the entire cube structure (with all 16 nodes alternating between red, green, and white; and all 128 channels alternating between blue and white), they experienced information overload. That is, too much was happening to discern what was going on at any given moment across the entire cube. As a result, viewers tended to focus on the center cube and ignore the outer cube.

Another problem relates to the inability to establish a consistent time frame reference in which to compare animation cases. This was partly due to the TESS program having to compete for CPU time slices with other CSC users, and partly due to the observed phenomenon that as the TESS program had less work to perform to generate the animation display (less going on in the animation), the animation executed more quickly.

The final comment was that they would prefer the addition of statistics graphics, such as bar or pie charts, showing node and channel utilizations as the animations executed. TESS does provide for the collection and display of such data.

4.2.2.2 Electrical and Computer Engineering. The animations seem to reflect actual hypercube processing, which provides a measure of validity to the underlying simulation model. The animations were useful in explaining the concepts of the imbalance and locality metrics and highlighted the impact of internodal communications overhead. Information overload was considered a problem and the time frame reference problem was also noted. Overall, it was concluded that animation shows great promise for other hypercube applications such as program tuning and program verification.

4.2.2.3 Personal Comments. As mentioned by both observation groups, the animations are useful for visually explaining the concepts of imbalance and locality, and graphically showing the impact of communications. However, the animations are most useful as a validation tool for the underlying simulation model. Watching the animations execute and being able to verify that that is how the architecture being modeled behaves, provides credibility for the model.

Both groups' remarks considering information overload and the time frame reference problems are valid concerns. For animations of this type to be useful when comparing different cases, they should be executed on a dedicated or single user system in order to avoid competition for CPU time with other users.

The comment by the Operations Research department members about wanting to see statistics graphics for node and channel utilizations is interesting. The request is driven by their knowledge of the use of TESS in more conventional applications, such as the animation of a factory or assembly line. In these applications, utilization statistics are important and displayed in various graphical forms. It is their knowledge of these types of applications and their expectancy to see utilization graphics that drives their request to see them in this rather unconventional application of TESS in which the only measure of concern is the time to complete the job.

4.2.3 Evaluation of Second Research Hypothesis. Due to the subjective nature of this portion of the thesis, this research hypothesis can neither be soundly rejected or accepted. Rather, based on personal opinion and the opinions of knowledgeable individuals who observed the animations it has been established that the animations do provide additional insight into the problem that could not be discerned from the textual output of the discrete event simulation. Unfortunately, these additional insights are difficult to quantify but include better understanding of the problem domain, better understanding of the impact of internodal IO overhead, and validation of the underlying simulation model.

5. Conclusions and Recommendations

5.1 Summary

It is apparent that load imbalance severely impacts a parallel processor's performance. The adverse effects are acute when even minor aberrations from a balanced load are allowed. The effect of load locality is minor and enters the speedup model primarily as an interactive term. This would suggest that locality effects, though minor, influence speedup behavior in ways that depend on the degree of imbalance. The intensity of IO is significant and affects the speedup across all levels of locality and imbalance.

The more IO involved in a process compared to CPU processing, the worse the speedup characteristics. This is intuitive since IO preempts node processing and introduces overhead which a single processor would not experience. What is not intuitive is that the IO load does not interact with the other terms. Apparently, higher IO loads cause a consistent worsening of performance regardless of the imbalance or locality of the load.

The findings of this research have serious impact on algorithm decomposition strategies. Given a known CPU to IO load, the balanced case speedup can be determined by simulation or benchmarking. As soon as processor imbalance is allowed, dramatic performance degradation can result. This research indicates that imbalance could not be overcome by locality. However, the affinity one node might have for another in terms of its IO was not modeled. If such an affinity were known, it is predicted that intelligent spatial loading, even if unbalanced, would be useful. However, the simple relocation of unbalanced loads may not recover the inherent loss of speedup caused by the unbalanced condition.

The use of the imbalance metric (B) and the locality metric (L) are simple statistics which can be used to model any process which can be monitored during

execution. Simulation allows a statistical approach to predicting process performance which provides a convenient framework for analysis. Sensitivity analysis is possible with multiple simulation runs.

The animated simulation cases showed that, though subjective, being able to "watch" the dynamic nature of load imbalance, locality, and IO intensity provided additional insight into the problem. One particular strong point of the animation is its use as a validation tool for the underlying simulation model.

5.2 *Recommendations for Future Research*

Several issues remain to be investigated. First, what happens when the messages generated by a node must be sent to all other nodes? Clearly, this situation will worsen the IO load and may change the interpretation of the analysis. Second, does the dimension of the hypercube affect the performance as the load becomes unbalanced? That is, would load imbalance on an 8 node or 64 node machine be similar to the 16 node case? It is suspected that the initial cube dimension will have an effect such that the lower dimensioned cubes are more adversely affected. However, this conjecture is made with caution since experience has indicated counter-intuitive results. Third, what are the affects of load imbalance and locality on speedup when process affinity with respect to IO is considered? Fourth, what are the affects of load imbalance on speedup across various parallel processor architectures and interconnection networks? This thesis limited the architecture to the hypercube structure: would imbalance have the same effect on a ring or tree architecture as on the hypercube architecture? Finally, it is suggested that animated simulation be used as a means of gaining answers to the above questions; provided a dedicated animation workstation is available.

Appendix A. *Data Set, SAS Program, and Regression Results from Message Transmission Analysis*

SAS Program and Embedded Data Set

```
OPTIONS LINESIZE=72;
TITLE 'COMMUNICATION TIME FUNCTION';
DATA TIMES;
INPUT INTNODES LENGTH TIME;
HOPS = INTNODES + 1;
XMISSION = HOPS * LENGTH;
CARDS;
 0  5   1.125000
 0  5   1.125000
 0  150   1.225000
 0  150   1.225000
 0  300   1.575000
 0  300   1.575000
 0  450   1.575000
 0  450   1.600000
 0  600   1.850000
 0  600   1.850000
 0  750   2.000000
 0  750   1.875000
 0  900   2.175000
 0  900   2.150000
 0  1024   2.300000
```

0	1024	2.275000
0	100	1.300000
0	200	1.275000
0	700	1.825000
0	800	1.925000
0	5	1.125000
0	5	1.125000
0	150	1.250000
0	150	1.400000
0	300	1.450000
0	300	1.450000
0	450	1.575000
0	450	1.575000
0	600	1.925000
0	600	1.900000
0	750	1.875000
0	750	1.875000
0	900	2.050000
0	900	2.025000
0	1024	2.350000
0	1024	2.325000
0	100	1.175000
0	200	1.450000
0	700	2.025000
0	800	1.925000
0	5	1.125000
0	5	1.125000
0	150	1.250000

0	150	1.225000
0	300	1.450000
0	300	1.675000
0	450	1.825000
0	450	1.600000
0	600	1.950000
0	600	1.925000
0	750	1.875000
0	750	1.875000
0	900	2.275000
0	900	2.225000
0	1024	2.375000
0	1024	2.425000
0	100	1.200000
0	200	1.250000
0	700	1.825000
0	800	1.925000
0	5	1.125000
0	5	1.125000
0	150	1.525000
0	150	1.250000
0	300	1.700000
0	300	1.450000
0	450	1.600000
0	450	1.575000
0	600	2.025000
0	600	2.000000
0	750	1.900000

0	750	1.875000
0	900	2.325000
0	900	2.025000
0	1024	2.175000
0	1024	2.475000
0	100	1.200000
0	200	1.550000
0	700	1.825000
0	800	1.950000
0	5	1.125000
0	5	1.125000
0	150	1.225000
0	150	1.225000
0	300	1.450000
0	300	1.775000
0	450	1.575000
0	450	1.575000
0	600	2.050000
0	600	2.050000
0	750	1.875000
0	750	1.900000
0	900	2.350000
0	900	2.350000
0	1024	2.475000
0	1024	2.475000
0	100	1.500000
0	200	1.625000
0	700	2.125000

0	800	2.250000
1	5	1.650000
1	5	1.625000
1	150	1.925000
1	150	1.900000
1	300	2.200000
1	300	2.200000
1	450	2.475000
1	450	2.475000
1	600	2.750000
1	600	2.750000
1	750	3.025000
1	750	3.025000
1	900	3.325000
1	900	3.300000
1	1024	3.550000
1	1024	3.525000
1	100	1.825000
1	200	2.000000
1	700	2.925000
1	800	3.125000
1	1	1.650000
1	1	1.650000
1	150	1.925000
1	150	1.925000
1	300	2.300000
1	300	2.200000
1	450	2.475000

1	450	2.475000
1	600	2.750000
1	600	2.750000
1	750	3.025000
1	750	3.025000
1	900	3.300000
1	900	3.325000
1	1024	3.525000
1	1024	3.550000
1	100	1.825000
1	200	2.000000
1	700	2.925000
1	800	3.125000
1	5	1.650000
1	5	1.625000
1	150	1.925000
1	150	1.925000
1	300	2.200000
1	300	2.200000
1	450	2.500000
1	450	2.500000
1	600	2.750000
1	600	2.750000
1	750	3.025000
1	750	3.025000
1	900	3.325000
1	900	3.300000
1	1024	3.550000

1	1024	3.525000
1	100	1.825000
1	200	2.000000
1	700	2.950000
1	800	3.125000
1	5	1.650000
1	5	1.650000
1	150	1.900000
1	150	1.925000
1	300	2.200000
1	300	2.200000
1	450	2.600000
1	450	2.775000
1	600	2.750000
1	600	2.750000
1	750	3.025000
1	750	3.025000
1	900	3.300000
1	900	3.600000
1	1024	3.550000
1	1024	3.525000
1	100	1.825000
1	200	2.000000
1	700	2.950000
1	800	3.125000
1	5	1.650000
1	5	1.650000
1	150	1.925000

1	150	1.925000
1	300	2.200000
1	300	2.200000
1	450	2.650000
1	450	2.475000
1	600	2.750000
1	600	2.750000
1	750	3.025000
1	750	3.025000
1	900	3.325000
1	900	3.600000
1	1024	3.550000
1	1024	3.550000
1	100	1.825000
1	200	2.000000
1	700	2.925000
1	800	3.125000
1	5	1.650000
1	5	1.650000
1	150	1.900000
1	150	1.925000
1	300	2.200000
1	300	2.200000
1	450	2.475000
1	450	2.500000
1	600	2.750000
1	600	2.750000
1	750	3.050000

1	750	3.025000
1	900	3.300000
1	900	3.300000
1	1024	3.550000
1	1024	3.550000
1	100	1.825000
1	200	2.000000
1	700	2.925000
1	800	3.125000
1	5	1.650000
1	5	1.650000
1	150	1.900000
1	150	1.925000
1	300	2.175000
1	300	2.200000
1	450	2.500000
1	450	2.825000
1	600	2.750000
1	600	2.750000
1	750	3.025000
1	750	3.025000
1	900	3.325000
1	900	3.300000
1	1024	3.550000
1	1024	3.525000
1	100	1.825000
1	200	2.000000
1	700	2.925000

1	800	3.125000
1	5	1.650000
1	5	1.650000
1	150	1.925000
1	150	1.900000
1	300	2.200000
1	300	2.200000
1	450	2.825000
1	450	2.800000
1	600	2.750000
1	600	2.750000
1	750	3.025000
1	750	3.025000
1	900	3.650000
1	900	3.300000
1	1024	3.550000
1	1024	3.550000
1	100	1.825000
1	200	2.000000
1	700	2.925000
1	800	3.125000
1	5	1.650000
1	5	1.650000
1	150	1.900000
1	150	1.925000
1	300	2.175000
1	300	2.200000
1	450	2.475000

1	450	2.825000
1	600	3.100000
1	600	3.075000
1	750	3.025000
1	750	3.025000
1	900	3.300000
1	900	3.325000
1	1024	3.525000
1	1024	3.550000
1	100	1.825000
1	200	2.000000
1	700	2.925000
1	800	3.125000
1	5	1.650000
1	5	1.650000
1	150	1.925000
1	150	1.925000
1	300	2.200000
1	300	2.200000
1	450	2.825000
1	450	2.475000
1	600	2.750000
1	600	2.750000
1	750	3.025000
1	750	3.025000
1	900	3.300000
1	900	3.325000
1	1024	3.550000

1	1024	3.525000
1	100	1.825000
1	200	2.000000
1	700	3.250000
1	800	3.125000
2	5	2.200000
2	5	2.200000
2	150	2.575000
2	150	2.575000
2	300	2.975000
2	300	3.000000
2	450	3.400000
2	450	3.400000
2	600	3.775000
2	600	3.775000
2	750	4.300000
2	750	4.275000
2	900	4.575000
2	900	4.575000
2	1024	4.925000
2	1024	4.900000
2	100	2.450000
2	200	2.700000
2	700	4.050000
2	800	4.300000
2	5	2.200000
2	5	2.200000
2	150	2.575000

2	150	2.600000
2	300	2.975000
2	300	2.975000
2	450	3.400000
2	450	3.375000
2	600	3.775000
2	600	3.775000
2	750	4.175000
2	750	4.275000
2	900	4.575000
2	900	4.575000
2	1024	4.925000
2	1024	4.900000
2	100	2.450000
2	200	2.725000
2	700	4.150000
2	800	4.400000
2	5	2.200000
2	5	2.200000
2	150	2.600000
2	150	2.575000
2	300	2.975000
2	300	3.000000
2	450	3.400000
2	450	3.375000
2	600	3.775000
2	600	3.775000
2	750	4.275000

2	750	4.175000
2	900	4.575000
2	900	4.575000
2	1024	4.900000
2	1024	4.900000
2	100	2.450000
2	200	2.700000
2	700	4.050000
2	800	4.425000
2	5	2.200000
2	5	2.200000
2	150	2.575000
2	150	2.600000
2	300	2.975000
2	300	2.975000
2	450	3.400000
2	450	3.375000
2	600	3.775000
2	600	3.775000
2	750	4.175000
2	750	4.350000
2	900	4.575000
2	900	4.750000
2	1024	4.900000
2	1024	4.925000
2	100	2.450000
2	200	2.700000
2	700	4.050000

2	800	4.300000
2	5	2.200000
2	5	2.200000
2	150	2.575000
2	150	2.575000
2	300	2.975000
2	300	2.975000
2	450	3.400000
2	450	3.375000
2	600	3.775000
2	600	3.775000
2	750	4.300000
2	750	4.175000
2	900	4.575000
2	900	4.575000
2	1024	4.900000
2	1024	4.925000
2	100	2.450000
2	200	2.700000
2	700	4.025000
2	800	4.325000
2	5	2.200000
2	5	2.200000
2	150	2.600000
2	150	2.900000
2	300	2.975000
2	300	3.000000
2	450	3.375000

2	450	3.375000
2	600	3.775000
2	600	3.800000
2	750	4.275000
2	750	4.175000
2	900	4.575000
2	900	4.575000
2	1024	4.900000
2	1024	5.025000
2	100	2.450000
2	200	2.700000
2	700	4.025000
2	800	4.325000
2	5	2.200000
2	5	2.200000
2	150	2.575000
2	150	2.925000
2	300	2.975000
2	300	3.325000
2	450	3.375000
2	450	3.400000
2	600	3.775000
2	600	3.775000
2	750	4.350000
2	750	4.325000
2	900	4.575000
2	900	4.575000
2	1024	4.900000

2	1024	4.900000
2	100	2.475000
2	200	2.700000
2	700	4.200000
2	800	4.325000
2	5	2.200000
2	5	2.200000
2	150	2.575000
2	150	2.925000
2	300	2.975000
2	300	3.000000
2	450	3.400000
2	450	3.375000
2	600	3.775000
2	600	3.775000
2	750	4.275000
2	750	4.275000
2	900	4.575000
2	900	4.575000
2	1024	4.900000
2	1024	4.900000
2	100	2.450000
2	200	2.725000
2	700	4.025000
2	800	4.325000
2	5	2.200000
2	5	2.200000
2	150	2.750000

2	150	2.575000
2	300	2.975000
2	300	3.000000
2	450	3.400000
2	450	3.375000
2	600	3.775000
2	600	3.775000
2	750	4.325000
2	750	4.325000
2	900	4.575000
2	900	4.575000
2	1024	4.925000
2	1024	4.900000
2	100	2.450000
2	200	2.725000
2	700	4.050000
2	800	4.300000
2	5	2.200000
2	5	2.200000
2	150	2.575000
2	150	2.925000
2	300	2.975000
2	300	3.300000
2	450	3.400000
2	450	3.375000
2	600	3.775000
2	600	3.775000
2	750	4.175000

2	750	4.175000
2	900	4.575000
2	900	4.575000
2	1024	4.925000
2	1024	4.900000
2	100	2.450000
2	200	2.725000
2	700	4.250000
2	800	4.300000
3	5	2.750000
3	5	2.750000
3	150	3.250000
3	150	3.250000
3	300	3.750000
3	300	3.775000
3	450	4.300000
3	450	4.400000
3	600	4.800000
3	600	4.825000
3	750	5.600000
3	750	5.600000
3	900	5.850000
3	900	5.825000
3	1024	6.275000
3	1024	6.275000
3	100	3.100000
3	200	3.400000
3	700	5.450000

3	800	5.775000
3	5	2.750000
3	5	2.750000
3	150	3.250000
3	150	3.250000
3	300	3.750000
3	300	3.750000
3	450	4.275000
3	450	4.300000
3	600	4.800000
3	600	4.800000
3	750	5.325000
3	750	5.650000
3	900	5.850000
3	900	5.825000
3	1024	6.275000
3	1024	6.275000
3	100	3.075000
3	200	3.425000
3	700	5.475000
3	800	5.500000
3	5	2.750000
3	5	2.750000
3	150	3.250000
3	150	3.250000
3	300	3.750000
3	300	3.775000
3	450	4.300000

3	450	4.300000
3	600	4.800000
3	600	4.800000
3	750	5.675000
3	750	5.425000
3	900	5.850000
3	900	5.850000
3	1024	6.275000
3	1024	6.250000
3	100	3.100000
3	200	3.400000
3	700	5.500000
3	800	5.500000
3	5	2.775000
3	5	2.750000
3	150	3.250000
3	150	3.250000
3	300	3.775000
3	300	3.750000
3	450	4.300000
3	450	4.275000
3	600	4.825000
3	600	4.800000
3	750	5.650000
3	750	5.650000
3	900	5.850000
3	900	5.850000
3	1024	6.275000

3	1024	6.275000
3	100	3.075000
3	200	3.400000
3	700	5.475000
3	800	5.500000
3	5	2.750000
3	5	2.750000
3	150	3.250000
3	150	3.250000
3	300	3.775000
3	300	3.750000
3	450	4.275000
3	450	4.300000
3	600	4.800000
3	600	4.800000
3	750	5.675000
3	750	5.475000
3	900	5.850000
3	900	5.850000
3	1024	6.250000
3	1024	6.275000
3	100	3.075000
3	200	3.425000
3	700	5.500000
3	800	5.500000
4	5	3.300000
4	5	3.300000
4	150	3.925000

```
4  150  3.925000
4  300  4.550000
4  300  4.550000
4  450  5.525000
4  450  5.175000
4  600  5.850000
4  600  5.825000
4  750  6.475000
4  750  6.800000
4  900  7.125000
4  900  7.100000
4  1024  7.650000
4  1024  7.625000
4  100  3.700000
4  200  4.125000
4  700  6.250000
4  800  6.675000
;
PROC SORT;
  BY LENGTH;
PROC PLOT;
  PLOT TIME*LENGTH;
PROC REG;
  MODEL TIME=XMISSION INTNODES;
RUN;
```

REGRESSION RESULTS

DEP VARIABLE: TIME

ANALYSIS OF VARIANCE

SOURCE	DF	SUM OF SQUARES	MEAN SQUARE	F VALUE	PROB>F
MODEL	2	1020.69887	510.34943592	50304.956	0.0001
ERROR	617	6.25953440	0.01014511		
C TOTAL	619	1026.95841			

ROOT MSE	0.1007229	R-SQUARE	0.9939
DEP MEAN	3.17375	ADJ R-SQ	0.9939
C.V.	3.173626		

PARAMETER ESTIMATES

VARIABLE	DF	PARAMETER ESTIMATE	STANDARD ERROR	T FOR H0: PARAMETER=0	PROB > |T|
INTERCEP	1	1.23168850	0.007685447	160.262	0.0001
XMISSION	1	0.0008968224	.00000437042	205.203	0.0001
INTNODES	1	0.48499870	0.004477457	108.320	0.0001

Appendix B. *SLAM II Source Code*

```
GEN,MOORE,NEW,07/28/87,1,YES,YES,YES/YES,YES,YES/1,72;
LIMITS,97,10,256;
SEEDS,43676651(1),6121137(2),9431826(3);

ARRAY(1,24)/0,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,2,3,5,9,34,35,36,37;
ARRAY(2,24)/1,0,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,4,6,10,38,39,40,41;
ARRAY(3,24)/2,1,0,1,2,1,3,1,2,1,4,1,2,1,3,1,4,1,7,11,43,42,44,45;
ARRAY(4,24)/1,2,1,0,1,2,1,3,1,2,1,4,1,2,1,3,3,2,8,12,47,46,48,49;
ARRAY(5,24)/3,1,2,1,0,1,2,1,3,1,2,1,4,1,2,1,6,7,1,13,51,52,50,53;
ARRAY(6,24)/1,3,1,2,1,0,1,2,1,3,1,2,1,4,1,2,5,8,2,14,55,56,54,57;
ARRAY(7,24)/2,1,3,1,2,1,0,1,2,1,3,1,2,1,4,1,8,5,3,15,60,59,58,61;
ARRAY(8,24)/1,2,1,3,1,2,1,0,1,2,1,3,1,2,1,4,7,6,4,16,64,63,62,65;
ARRAY(9,24)/4,1,2,1,3,1,2,1,0,1,2,1,3,1,2,1,10,11,13,1,67,68,69,66;
ARRAY(10,24)/1,4,1,2,1,3,1,2,1,0,1,2,1,3,1,2,9,12,14,2,71,72,73,70;
ARRAY(11,24)/2,1,4,1,2,1,3,1,2,1,0,1,2,1,3,1,12,9,15,3,76,75,77,74;
ARRAY(12,24)/1,2,1,4,1,2,1,3,1,2,1,0,1,2,1,3,11,10,16,4,80,79,81,78;
ARRAY(13,24)/3,1,2,1,4,1,2,1,3,1,2,1,0,1,2,1,14,15,9,5,84,85,83,82;
ARRAY(14,24)/1,3,1,2,1,4,1,2,1,3,1,2,1,0,1,2,13,16,10,6,88,89,87,86;
ARRAY(15,24)/2,1,3,1,2,1,4,1,2,1,3,1,2,1,0,1,16,13,11,7,93,92,91,90;
ARRAY(16,24)/1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,0,15,14,12,8,97,96,95,94;

NETWORK;
        RESOURCE/1,NODE1(1),1,17;
        RESOURCE/2,NODE2(1),2,18;
        RESOURCE/3,NODE3(1),3,19;
```

```
RESOURCE/4,NODE4(1),4,20;
RESOURCE/5,NODE5(1),5,21;
RESOURCE/6,NODE6(1),6,22;
RESOURCE/7,NODE7(1),7,23;
RESOURCE/8,NODE8(1),8,24;
RESOURCE/9,NODE9(1),9,25;
RESOURCE/10,NODE10(1),10,26;
RESOURCE/11,NODE11(1),11,27;
RESOURCE/12,NODE12(1),12,28;
RESOURCE/13,NODE13(1),13,29;
RESOURCE/14,NODE14(1),14,30;
RESOURCE/15,NODE15(1),15,31;
RESOURCE/16,NODE16(1),16,32;
RESOURCE/17,CUBE(1),33;

;  FILE NUMBERS FOR CHANNEL QUEUES.

;    C1X2     34
;       3     35
;       5     36
;       9     37

;    C2X1     38
;       4     39
;       6     40
;      10     41

;    C3X1     42
```

```
;       4      43
;       7      44
;      11      45

;    C4X2      46
;       3      47
;       8      48
;      12      49

;    C5X1      50
;       6      51
;       7      52
;      13      53

;    C6X2      54
;       5      55
;       8      56
;      14      57

;    C7X3      58
;       5      59
;       8      60
;      15      61

;    C8X4      62
;       6      63
;       7      64
;      16      65
```

```
;    C9X1     66
;      10     67
;      11     68
;      13     69

;   C10X2     70
;       9     71
;      12     72
;      14     73

;   C11X3     74
;       9     75
;      12     76
;      15     77

;   C12X4     78
;      10     79
;      11     80
;      16     81

;   C13X5     82
;       9     83
;      14     84
;      15     85

;   C14X6     86
;      10     87
```

```
;        13        88
;        16        89

;     C15X7        90
;        11        91
;        13        92
;        16        93

;     C16X8        94
;        12        95
;        14        96
;        15        97



        CREATE,400,0,1,1,1;     A new job enters the system.
        AWAIT(33),CUBE/1,,1;    Get exclusive control of the cube.
        ASSIGN,XX(1)=0,
               XX(2)=.6158445,
               XX(3)=.0008968224,
               XX(4)=.4849987,1; XX(1) =  NUMBER OF ENTITIES ACTIVE.
;                        XX(2) =  SENDING & RECEIVING MSG OVERHEAD.
;                        XX(3) =  MS/BYTE XMISSION TIME.
;                        XX(4) =  INTERMEDIATE NODE OVERHEAD.

        ASSIGN,ATRIB(1)=TNOW,1;   Set job start time in ATRIB(1).
        ASSIGN,XX(1)=16,1;
```

```
        GOON,16;            PARTITION JOB INTO PARALLEL PROCESSES.
        ACTIVITY,,,ND1;
        ACTIVITY,,,ND2;
        ACTIVITY,,,ND3;
        ACTIVITY,,,ND4;
        ACTIVITY,,,ND5;
        ACTIVITY,,,ND6;
        ACTIVITY,,,ND7;
        ACTIVITY,,,ND8;
        ACTIVITY,,,ND9;
        ACTIVITY,,,ND10;
        ACTIVITY,,,ND11;
        ACTIVITY,,,ND12;
        ACTIVITY,,,ND13;
        ACTIVITY,,,ND14;
        ACTIVITY,,,ND15;
        ACTIVITY,,,ND16;

;       Assign Node id, # of elements, and length of each burst.
ND1     GOON,1;
        ASSIGN,ATRIB(2)=1,ATRIB(3)=16,1;
        ACTIVITY,,,WTND;

ND2     GOON,1;
        ASSIGN,ATRIB(2)=2,ATRIB(3)=16,1;
        ACTIVITY,,,WTND;
```

```
ND3     GOON,1;
        ASSIGN,ATRIB(2)=3,ATRIB(3)=16,1;
        ACTIVITY,,,WTND;

ND4     GOON,1;
        ASSIGN,ATRIB(2)=4,ATRIB(3)=16,1;
        ACTIVITY,,,WTND;

ND5     GOON,1;
        ASSIGN,ATRIB(2)=5,ATRIB(3)=16,1;
        ACTIVITY,,,WTND;

ND6     GOON,1;
        ASSIGN,ATRIB(2)=6,ATRIB(3)=16,1;
        ACTIVITY,,,WTND;

ND7     GOON,1;
        ASSIGN,ATRIB(2)=7,ATRIB(3)=16,1;
        ACTIVITY,,,WTND;

ND8     GOON,1;
        ASSIGN,ATRIB(2)=8,ATRIB(3)=16,1;
        ACTIVITY,,,WTND;

ND9     GOON,1;
        ASSIGN,ATRIB(2)=9,ATRIB(3)=16,1;
        ACTIVITY,,,WTND;
```

```
ND10  GOON,1;
      ASSIGN,ATRIB(2)=10,ATRIB(3)=16,1;
      ACTIVITY,,,WTND;

ND11  GOON,1;
      ASSIGN,ATRIB(2)=11,ATRIB(3)=16,1;
      ACTIVITY,,,WTND;

ND12  GOON,1;
      ASSIGN,ATRIB(2)=12,ATRIB(3)=16,1;
      ACTIVITY,,,WTND;

ND13  GOON,1;
      ASSIGN,ATRIB(2)=13,ATRIB(3)=16,1;
      ACTIVITY,,,WTND;

ND14  GOON,1;
      ASSIGN,ATRIB(2)=14,ATRIB(3)=16,1;
      ACTIVITY,,,WTND;

ND15  GOON,1;
      ASSIGN,ATRIB(2)=15,ATRIB(3)=16,1;
      ACTIVITY,,,WTND;

ND16  GOON,1;
      ASSIGN,ATRIB(2)=16,ATRIB(3)=16,1;
      ACTIVITY,,,WTND;
```

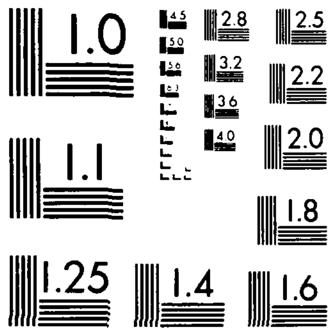
```
CHND  GOON,1;
      ACTIVITY,,ATRIB(3).EQ.0,DECN; ALL BURSTS FOR NODE ARE DONE.
      ACTIVITY,,ATRIB(3).GT.0,WTND; NOT DONE.

WTND  GOON,1;

      ASSIGN,ATRIB(4)=EXPON(3.228,1),1; ASSIGN BURST DURATION.

      AWAIT(ATRIB(2)=1,16),ATRIB(2)/1,,1;  WAIT FOR CORRECT NODE.
      ACTIVITY/ATRIB(2)=1,16,ATRIB(4)+XX(2); NODE BURST + MSG.

      FREE,ATRIB(2)/1,1;   Free the node that just processed.

      GOON,1;
      ASSIGN,XX(1)=XX(1)+1,2; INCR NUMBER OF ACTIVE ENTITIES.
      ACTIVITY,,,RTMG;          SEND ONE ENTITY AS A MESSAGE.
      ACTIVITY,,,DCNT;  SEND JOB ENTITY TO DECREMENT BURST COUNT.

DCNT  GOON,1;   DECREMENT THE NUMBER OF BURSTS FOR THIS NODE.
      ASSIGN,ATRIB(3)=ATRIB(3)-1,1;
      ACTIVITY,,,CHND; BRANCH BACK UP TO EXECUTE ANOTHER BURST.

DECN  GOON,1;   THIS NODE HAS COMPLETED ALL ITS BURSTS.
      ASSIGN,XX(1)=XX(1)-1,1;   DEC # OF ACTIVE ENTITIES BY ONE.
      ACTIVITY,,XX(1).EQ.0,DONE;  EVERYTHING DONE.
      ACTIVITY,,XX(1).GT.0,NDTM;
NDTM  TERMINATE;               TERMINATE THIS ENTITY.
```

```
;  SEND THE RESULTS OF THIS NODE BURST TO A RANDOM RECIPIENT.
RTMG  GOON,1;
      ASSIGN,II=UNFRM(1,16,2),
            ATRIB(3)=II,1;
      ASSIGN,ATRIB(10)=UNFRM(100,1024,3),1; ASN MSG PACKET SIZE.

      ACTIVITY,,ATRIB(3).EQ.ATRIB(2),RTMG; IF RCVR SAME AS SNDR,
      ACTIVITY,,ATRIB(3).NE.ATRIB(2),GTCH; PICK ANOTHER RCVR.


ASN   GOON,1;    Assign the next node and channel.
      ASSIGN,ATRIB(9)=ATRIB(2)+80,1;
      ACTIVITY,,ATRIB(2).EQ.ATRIB(3),S1;
      ACTIVITY,,ATRIB(2).NE.ATRIB(3),S2;
S2    ASSIGN,XX(5)=XX(4),1;    Intermediate processing time.
      ACTIVITY,,,RND;
S1    ASSIGN,XX(5)=XX(2),1;    Destination processing time.
      ACTIVITY,,,RND;

RND   GOON,1;
      ACTIVITY/ATRIB(9)=81,96,XX(5);  Receiving node overhead.

S3    GOON,1;
      ACTIVITY,,ATRIB(2).EQ.ATRIB(3),DEST;  DESTINATION NODE.
      ACTIVITY,,ATRIB(2).NE.ATRIB(3),FRND;  INTERMEDIATE NODE.

FRND  FREE,ATRIB(2),1;   FREE THE INTERMEDIATE NODE.
```

```
GTCH  ASSIGN,ATRIB(6)=ARRAY(ATRIB(2),ATRIB(3)),1; Get Column number.
GTND  ASSIGN,ATRIB(6)=ATRIB(6)+16,1;        Offset for node.
      ASSIGN,ATRIB(4)=ARRAY(ATRIB(2),ATRIB(6)),1; Get next node.
      ASSIGN,ATRIB(6)=ATRIB(6)+4,1;         Offset for channel.
      ASSIGN,ATRIB(7)=ARRAY(ATRIB(2),ATRIB(6)),1;  Get channel.


;                  WAIT FOR THE APPROPRIATE CHANNEL.

WTCH  GOON,1;
      ASSIGN,ATRIB(9)=ATRIB(7)-17;
      QUEUE(ATRIB(7)=34,97);
      ACTIVITY(1)/ATRIB(9)=17,80,XX(3)*ATRIB(10);

      GOON,1;
      ASSIGN,ATRIB(5)=ATRIB(4)+16,
             ATRIB(8)=ATRIB(4),1;

      PREEMPT(ATRIB(5)=17,32),ATRIB(8),,,1; PREEMPT THE
;                RECEIVING NODE SO IT CAN PROCESS TH

      ASSIGN,ATRIB(2)=ATRIB(4),1;

      ACTIVITY,,,ASN;          ASSIGN


DEST  GOON,1;        THIS I
      FREE,ATRIB
```

AD-A189 573 A SIMULATION STUDY OF A PARALLEL PROCESSOR WITH UNBALANCED LOADS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING 2/2

UNCLASSIFIED T S MOORE DEC 87 AFIT/GCS/ENG/87D-28 F/G 12/6 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
        ACTIVITY,,XX(1).GT.1,NTDN; MORE THAN 1 ENTY IS STILL ACTIVE.
        ACTIVITY,,XX(1).EQ.1,DONE;  THIS IS THE ONLY ENTITY
;                                                  STILL ACTIVE.
NTDN    GOON,1;
        ASSIGN,XX(1)=XX(1)-1,1; REMOVE THIS ENTITY FROM THE COUNT.
        TERMINATE;              TERMINATE THIS ENTITY.

DONE    GOON,1;     THIS IS THE LAST ENTITY IN THE SYSTEM.
        COLCT,INT(1),TIME IN SYSTEM,,1;  JOB IS FINISHED.
        FREE,CUBE/1,1;                            FREE THE CUBE.
        TERMINATE;   TERMINATE THE LAST ENTITY.

        END;
INIT,0,8000;
FIN;
```

Appendix C. *TESS History File*

The following commands were inserted into the SLAM II source code.

```
SCENARIO,GENBAL;

DOEVENT,HISTORY,TRACE OF CUBE,
ACT/S,1,96,
ACT/C,1,96;
```

The SCENARIO statement specifies the name of the TESS scenario. In this case the scenarion name is GENBAL.

The DOEVENT statement specifies a definition name and definition descriptor to be used in organizing the TESS storage of trace data. This particular statement causes event start and completion times to be stored for event numbers 1 thru 96.

The following pages represent the first 6.42 milliseconds of a 105 millisecond simulation job. Event types 1 and 2 correspond to activity start and stop, respectively.

****

EVENTTYPE	EVENTNUM	ACTDUR	TNOW
0.100000E+01	0.100000E+01	0.121888E+01	0.000000E+00
0.100000E+01	0.200000E+01	0.938787E+01	0.000000E+00
0.100000E+01	0.300000E+01	0.777158E+00	0.000000E+00
0.100000E+01	0.400000E+01	0.157421E+01	0.000000E+00
0.100000E+01	0.500000E+01	0.180110E+01	0.000000E+00
0.100000E+01	0.600000E+01	0.732536E+00	0.000000E+00
0.100000E+01	0.700000E+01	0.468113E+01	0.000000E+00
0.100000E+01	0.800000E+01	0.434609E+01	0.000000E+00
0.100000E+01	0.900000E+01	0.107520E+02	0.000000E+00
0.100000E+01	0.100000E+02	0.350823E+01	0.000000E+00
0.100000E+01	0.110000E+02	0.360363E+01	0.000000E+00
0.100000E+01	0.120000E+02	0.223002E+01	0.000000E+00
0.100000E+01	0.130000E+02	0.345290E+01	0.000000E+00
0.100000E+01	0.140000E+02	0.172168E+02	0.000000E+00
0.100000E+01	0.150000E+02	0.622873E+01	0.000000E+00
0.100000E+01	0.160000E+02	0.299228E+01	0.000000E+00
0.200000E+01	0.600000E+01	0.000000E+00	0.732536E+00
0.100000E+01	0.400000E+02	0.766668E+00	0.732536E+00
0.100000E+01	0.600000E+01	0.343768E+01	0.732536E+00

0.200000E+01	0.300000E+01	0.000000E+00	0.777158E+00
0.100000E+01	0.250000E+02	0.638033E+00	0.777158E+00
0.100000E+01	0.300000E+01	0.640220E+00	0.777158E+00
0.200000E+01	0.100000E+01	0.000000E+00	0.121888E+01
0.100000E+01	0.180000E+02	0.652530E+00	0.121888E+01
0.100000E+01	0.100000E+01	0.245333E+01	0.121888E+01
0.200000E+01	0.250000E+02	0.000000E+00	0.141519E+01
0.100000E+01	0.810000E+02	0.484999E+00	0.141519E+01
0.200000E+01	0.300000E+01	0.000000E+00	0.141738E+01
0.100000E+01	0.250000E+02	0.667948E+00	0.141738E+01
0.100000E+01	0.300000E+01	0.876256E+01	0.141738E+01
0.200000E+01	0.400000E+02	0.000000E+00	0.149920E+01
0.100000E+01	0.940000E+02	0.615845E+00	0.149920E+01
0.200000E+01	0.400000E+01	0.000000E+00	0.157421E+01
0.100000E+01	0.300000E+02	0.226832E+00	0.157421E+01
0.100000E+01	0.400000E+01	0.575054E+01	0.157421E+01
0.200000E+01	0.300000E+02	0.000000E+00	0.180104E+01
0.100000E+01	0.830000E+02	0.615844E+00	0.180104E+01
0.200000E+01	0.500000E+01	0.000000E+00	0.180110E+01
0.100000E+01	0.330000E+02	0.654347E+00	0.180110E+01
0.100000E+01	0.500000E+01	0.901651E+01	0.180110E+01
0.200000E+01	0.180000E+02	0.000000E+00	0.187141E+01
0.200000E+01	0.810000E+02	0.000000E+00	0.190019E+01
0.100000E+01	0.100000E+01	0.225702E+01	0.190019E+01
0.100000E+01	0.190000E+02	0.638033E+00	0.190019E+01
0.200000E+01	0.250000E+02	0.000000E+00	0.208533E+01
0.100000E+01	0.810000E+02	0.484999E+00	0.208533E+01
0.200000E+01	0.940000E+02	0.000000E+00	0.211505E+01

0.100000E+01	0.140000E+02	0.157176E+02	0.211505E+01
0.200000E+01	0.120000E+02	0.000000E+00	0.223002E+01
0.100000E+01	0.630000E+02	0.558628E+00	0.223002E+01
0.100000E+01	0.120000E+02	0.475731E+01	0.223002E+01
0.200000E+01	0.830000E+02	0.000000E+00	0.241689E+01
0.100000E+01	0.300000E+01	0.837889E+01	0.241689E+01
0.200000E+01	0.330000E+02	0.000000E+00	0.245545E+01
0.200000E+01	0.190000E+02	0.000000E+00	0.253822E+01
0.100000E+01	0.850000E+02	0.484999E+00	0.253822E+01
0.200000E+01	0.810000E+02	0.000000E+00	0.257032E+01
0.100000E+01	0.100000E+01	0.207188E+01	0.257032E+01
0.100000E+01	0.190000E+02	0.667948E+00	0.257032E+01
0.200000E+01	0.630000E+02	0.000000E+00	0.278865E+01
0.100000E+01	0.910000E+02	0.484999E+00	0.278865E+01
0.200000E+01	0.160000E+02	0.000000E+00	0.299228E+01
0.100000E+01	0.790000E+02	0.240145E+00	0.299228E+01
0.100000E+01	0.160000E+02	0.981307E+01	0.299228E+01
0.200000E+01	0.850000E+02	0.000000E+00	0.302322E+01
0.100000E+01	0.500000E+01	0.827939E+01	0.302322E+01
0.100000E+01	0.360000E+02	0.638033E+00	0.302322E+01
0.200000E+01	0.790000E+02	0.000000E+00	0.323243E+01
0.100000E+01	0.940000E+02	0.484999E+00	0.323243E+01
0.200000E+01	0.190000E+02	0.000000E+00	0.323827E+01
0.100000E+01	0.850000E+02	0.615844E+00	0.323827E+01
0.200000E+01	0.910000E+02	0.000000E+00	0.327364E+01
0.100000E+01	0.110000E+02	0.814937E+00	0.327364E+01
0.100000E+01	0.600000E+02	0.558628E+00	0.327364E+01
0.200000E+01	0.130000E+02	0.000000E+00	0.345290E+01

0.100000E+01	0.650000E+02	0.663650E+00	0.345290E+01
0.100000E+01	0.130000E+02	0.732528E+00	0.345290E+01
0.200000E+01	0.100000E+02	0.000000E+00	0.350823E+01
0.100000E+01	0.550000E+02	0.296040E+00	0.350823E+01
0.100000E+01	0.100000E+02	0.428902E+01	0.350823E+01
0.200000E+01	0.360000E+02	0.000000E+00	0.366125E+01
0.100000E+01	0.930000E+02	0.615845E+00	0.366125E+01
0.200000E+01	0.940000E+02	0.000000E+00	0.371743E+01
0.100000E+01	0.140000E+02	0.146002E+02	0.371743E+01
0.100000E+01	0.690000E+02	0.240145E+00	0.371743E+01
0.200000E+01	0.550000E+02	0.000000E+00	0.380427E+01
0.100000E+01	0.920000E+02	0.615845E+00	0.380427E+01
0.200000E+01	0.600000E+02	0.000000E+00	0.383227E+01
0.100000E+01	0.950000E+02	0.615845E+00	0.383227E+01
0.200000E+01	0.850000E+02	0.000000E+00	0.385412E+01
0.100000E+01	0.500000E+01	0.806434E+01	0.385412E+01
0.200000E+01	0.690000E+02	0.000000E+00	0.395757E+01
0.100000E+01	0.860000E+02	0.615844E+00	0.395757E+01
0.200000E+01	0.110000E+02	0.000000E+00	0.408863E+01
0.100000E+01	0.590000E+02	0.389917E+00	0.408863E+01
0.100000E+01	0.110000E+02	0.328163E+01	0.408863E+01
0.200000E+01	0.650000E+02	0.000000E+00	0.411655E+01
0.100000E+01	0.850000E+02	0.615845E+00	0.411655E+01
0.200000E+01	0.930000E+02	0.000000E+00	0.427710E+01
0.100000E+01	0.130000E+02	0.524172E+00	0.427710E+01
0.200000E+01	0.800000E+01	0.000000E+00	0.434609E+01
0.100000E+01	0.470000E+02	0.410812E+00	0.434609E+01
0.100000E+01	0.800000E+01	0.451727E+01	0.434609E+01

0.200000E+01	0.920000E+02	0.000000E+00	0.442011E+01
0.100000E+01	0.120000E+02	0.318306E+01	0.442011E+01
0.200000E+01	0.950000E+02	0.000000E+00	0.444812E+01
0.100000E+01	0.150000E+02	0.239646E+01	0.444812E+01
0.200000E+01	0.590000E+02	0.000000E+00	0.447855E+01
0.100000E+01	0.920000E+02	0.615845E+00	0.447855E+01
0.200000E+01	0.860000E+02	0.000000E+00	0.457342E+01
0.100000E+01	0.600000E+01	0.212639E+00	0.457342E+01
0.200000E+01	0.100000E+01	0.000000E+00	0.464220E+01
0.100000E+01	0.170000E+02	0.226842E+00	0.464220E+01
0.100000E+01	0.810000E+02	0.484999E+00	0.464220E+01
0.200000E+01	0.700000E+01	0.000000E+00	0.468113E+01
0.100000E+01	0.440000E+02	0.829577E+00	0.468113E+01
0.100000E+01	0.700000E+01	0.103569E+01	0.468113E+01
0.200000E+01	0.850000E+02	0.000000E+00	0.473239E+01
0.100000E+01	0.500000E+01	0.780191E+01	0.473239E+01
0.200000E+01	0.470000E+02	0.000000E+00	0.475690E+01
0.100000E+01	0.870000E+02	0.484999E+00	0.475690E+01
0.200000E+01	0.600000E+01	0.000000E+00	0.478606E+01
0.100000E+01	0.380000E+02	0.574785E+00	0.478606E+01
0.100000E+01	0.600000E+01	0.563328E+01	0.478606E+01
0.200000E+01	0.130000E+02	0.000000E+00	0.480127E+01
0.100000E+01	0.650000E+02	0.820376E+00	0.480127E+01
0.100000E+01	0.130000E+02	0.145021E+01	0.480127E+01
0.200000E+01	0.170000E+02	0.000000E+00	0.486905E+01
0.100000E+01	0.820000E+02	0.484999E+00	0.486905E+01
0.200000E+01	0.920000E+02	0.000000E+00	0.509439E+01
0.100000E+01	0.120000E+02	0.312462E+01	0.509439E+01

0.200000E+01	0.810000E+02	0.000000E+00	0.512720E+01
0.100000E+01	0.100000E+01	0.165211E+01	0.512720E+01
0.100000E+01	0.200000E+02	0.654347E+00	0.512720E+01
0.200000E+01	0.870000E+02	0.000000E+00	0.524190E+01
0.100000E+01	0.700000E+01	0.959918E+00	0.524190E+01
0.100000E+01	0.410000E+02	0.410812E+00	0.524190E+01
0.200000E+01	0.820000E+02	0.000000E+00	0.535404E+01
0.100000E+01	0.200000E+01	0.451882E+01	0.535404E+01
0.100000E+01	0.240000E+02	0.226842E+00	0.535404E+01
0.200000E+01	0.380000E+02	0.000000E+00	0.536084E+01
0.100000E+01	0.850000E+02	0.484999E+00	0.536084E+01
0.200000E+01	0.440000E+02	0.000000E+00	0.551071E+01
0.100000E+01	0.950000E+02	0.615845E+00	0.551071E+01
0.200000E+01	0.240000E+02	0.000000E+00	0.558089E+01
0.100000E+01	0.900000E+02	0.615845E+00	0.558089E+01
0.200000E+01	0.650000E+02	0.000000E+00	0.562165E+01
0.200000E+01	0.410000E+02	0.000000E+00	0.565272E+01
0.100000E+01	0.830000E+02	0.484999E+00	0.565272E+01
0.200000E+01	0.200000E+02	0.000000E+00	0.578155E+01
0.100000E+01	0.890000E+02	0.615845E+00	0.578155E+01
0.200000E+01	0.850000E+02	0.000000E+00	0.584584E+01
0.100000E+01	0.500000E+01	0.717346E+01	0.584584E+01
0.100000E+01	0.350000E+02	0.574785E+00	0.584584E+01
0.200000E+01	0.950000E+02	0.000000E+00	0.612656E+01
0.100000E+01	0.150000E+02	0.133386E+01	0.612656E+01
0.200000E+01	0.830000E+02	0.000000E+00	0.613771E+01
0.100000E+01	0.300000E+01	0.514306E+01	0.613771E+01
0.100000E+01	0.280000E+02	0.410812E+00	0.613771E+01

0.200000E+01	0.900000E+02	0.000000E+00	0.619673E+01
0.100000E+01	0.100000E+02	0.221637E+01	0.619673E+01
0.200000E+01	0.700000E+01	0.000000E+00	0.620182E+01
0.100000E+01	0.430000E+02	0.885662E+00	0.620182E+01
0.100000E+01	0.700000E+01	0.130333E+01	0.620182E+01
0.200000E+01	0.130000E+02	0.000000E+00	0.625148E+01
0.100000E+01	0.680000E+02	0.161787E+00	0.625148E+01
0.100000E+01	0.130000E+02	0.305453E+02	0.625148E+01
0.200000E+01	0.890000E+02	0.000000E+00	0.639739E+01
0.100000E+01	0.900000E+01	0.497042E+01	0.639739E+01
0.200000E+01	0.680000E+02	0.000000E+00	0.641327E+01
0.100000E+01	0.950000E+02	0.484999E+00	0.641327E+01
0.200000E+01	0.350000E+02	0.000000E+00	0.642063E+01

Appendix D. *TESS Rule Set*

```
REPORT OF RULES:    R4        TYPE:    SLAM
ZOOM:SIM/S,,OUT,3.5;
COLOR:ACT/S,1,N1,1,1,RED,;
COLOR:ACT/C,1,N1,1,1,WHITE,;
COLOR:ACT/S,81,N1,1,1,GREEN,;
COLOR:ACT/C,81,N1,1,1,WHITE,;
COLOR:ACT/S,2,N2,1,1,RED,;
COLOR:ACT/C,2,N2,1,1,WHITE,;
COLOR:ACT/S,82,N2,1,1,GREEN,;
COLOR:ACT/C,82,N2,1,1,WHITE,;
COLOR:ACT/S,3,N3,1,1,RED,;
COLOR:ACT/C,3,N3,1,1,WHITE,;
COLOR:ACT/S,83,N3,1,1,GREEN,;
COLOR:ACT/C,83,N3,1,1,WHITE,;
COLOR:ACT/S,4,N4,1,1,RED,;
COLOR:ACT/C,4,N4,1,1,WHITE,;
COLOR:ACT/S,84,N4,1,1,GREEN,;
COLOR:ACT/C,84,N4,1,1,WHITE,;
COLOR:ACT/S,5,N5,1,1,RED,;
COLOR:ACT/C,5,N5,1,1,WHITE,;
COLOR:ACT/S,85,N5,1,1,GREEN,;
COLOR:ACT/C,85,N5,1,1,WHITE,;
COLOR:ACT/S,6,N6,1,1,RED,;
COLOR:ACT/C,6,N6,1,1,WHITE,;
COLOR:ACT/S,86,N6,1,1,GREEN,;
COLOR:ACT/C,86,N6,1,1,WHITE,;
```

```
COLOR:ACT/S,7,N7,1,1,RED,;
COLOR:ACT/C,7,N7,1,1,WHITE,;
COLOR:ACT/S,87,N7,1,1,GREEN,;
COLOR:ACT/C,87,N7,1,1,WHITE,;
COLOR:ACT/S,8,N8,1,1,RED,;
COLOR:ACT/C,8,N8,1,1,WHITE,;
COLOR:ACT/S,88,N8,1,1,GREEN,;
COLOR:ACT/C,88,N8,1,1,WHITE,;
COLOR:ACT/S,9,N9,1,1,RED,;
COLOR:ACT/C,9,N9,1,1,WHITE,;
COLOR:ACT/S,89,N9,1,1,GREEN,;
COLOR:ACT/C,89,N9,1,1,WHITE,;
COLOR:ACT/S,10,N10,1,1,RED,;
COLOR:ACT/C,10,N10,1,1,WHITE,;
COLOR:ACT/S,90,N10,1,1,GREEN,;
COLOR:ACT/C,90,N10,1,1,WHITE,;
COLOR:ACT/S,11,N11,1,1,RED,;
COLOR:ACT/C,11,N11,1,1,WHITE,;
COLOR:ACT/S,91,N11,1,1,GREEN,;
COLOR:ACT/C,91,N11,1,1,WHITE,;
COLOR:ACT/S,12,N12,1,1,RED,;
COLOR:ACT/C,12,N12,1,1,WHITE,;
COLOR:ACT/S,92,N12,1,1,GREEN,;
COLOR:ACT/C,92,N12,1,1,WHITE,;
COLOR:ACT/S,13,N13,1,1,RED,;
COLOR:ACT/C,13,N13,1,1,WHITE,;
COLOR:ACT/S,93,N13,1,1,GREEN,;
COLOR:ACT/C,93,N13,1,1,WHITE,;
```

```
COLOR:ACT/S,14,N14,1,1,RED,;
COLOR:ACT/C,14,N14,1,1,WHITE,;
COLOR:ACT/S,94,N14,1,1,GREEN,;
COLOR:ACT/C,94,N14,1,1,WHITE,;
COLOR:ACT/S,15,N15,1,1,RED,;
COLOR:ACT/C,15,N15,1,1,WHITE,;
COLOR:ACT/S,95,N15,1,1,GREEN,;
COLOR:ACT/C,95,N15,1,1,WHITE,;
COLOR:ACT/S,16,N16,1,1,RED,;
COLOR:ACT/C,16,N16,1,1,WHITE,;
COLOR:ACT/S,96,N16,1,1,GREEN,;
COLOR:ACT/C,96,N16,1,1,WHITE,;
COLOR:ACT/S,17,C1X2,1,1,BLUE,;
COLOR:ACT/C,17,C1X2,1,1,WHITE,;
COLOR:ACT/S,18,C1X3,1,1,BLUE,;
COLOR:ACT/C,18,C1X3,1,1,WHITE,;
COLOR:ACT/S,19,C1X5,1,1,BLUE,;
COLOR:ACT/C,19,C1X5,1,1,WHITE,;
COLOR:ACT/S,20,C1X9,1,1,BLUE,;
COLOR:ACT/C,20,C1X9,1,1,WHITE,;
COLOR:ACT/S,21,C2X1,1,1,BLUE,;
COLOR:ACT/C,21,C2X1,1,1,WHITE,;
COLOR:ACT/S,22,C2X4,1,1,BLUE,;
COLOR:ACT/C,22,C2X4,1,1,WHITE,;
COLOR:ACT/S,23,C2X6,1,1,BLUE,;
COLOR:ACT/C,23,C2X6,1,1,WHITE,;
COLOR:ACT/S,24,C2X10,1,1,BLUE,;
COLOR:ACT/C,24,C2X10,1,1,WHITE,;
```

```
COLOR:ACT/S,25,C3X1,1,1,BLUE,;
COLOR:ACT/C,25,C3X1,1,1,WHITE,;
COLOR:ACT/S,26,C3X4,1,1,BLUE,;
COLOR:ACT/C,26,C3X4,1,1,WHITE,;
COLOR:ACT/S,27,C3X7,1,1,BLUE,;
COLOR:ACT/C,27,C3X7,1,1,WHITE,;
COLOR:ACT/S,28,C3X11,1,1,BLUE,;
COLOR:ACT/C,28,C3X11,1,1,WHITE,;
COLOR:ACT/S,29,C4X2,1,1,BLUE,;
COLOR:ACT/C,29,C4X2,1,1,WHITE,;
COLOR:ACT/S,30,C4X3,1,1,BLUE,;
COLOR:ACT/C,30,C4X3,1,1,WHITE,;
COLOR:ACT/S,31,C4X8,1,1,BLUE,;
COLOR:ACT/C,31,C4X8,1,1,WHITE,;
COLOR:ACT/S,32,C4X12,1,1,BLUE,;
COLOR:ACT/C,32,C4X12,1,1,WHITE,;
COLOR:ACT/S,33,C5X1,1,1,BLUE,;
COLOR:ACT/C,33,C5X1,1,1,WHITE,;
COLOR:ACT/S,34,C5X6,1,1,BLUE,;
COLOR:ACT/C,34,C5X6,1,1,WHITE,;
COLOR:ACT/S,35,C5X7,1,1,BLUE,;
COLOR:ACT/C,35,C5X7,1,1,WHITE,;
COLOR:ACT/S,36,C5X13,1,1,BLUE,;
COLOR:ACT/C,36,C5X13,1,1,WHITE,;
COLOR:ACT/S,37,C6X2,1,1,BLUE,;
COLOR:ACT/C,37,C6X2,1,1,WHITE,;
COLOR:ACT/S,38,C6X5,1,1,BLUE,;
COLOR:ACT/C,38,C6X5,1,1,WHITE,;
```

```
COLOR:ACT/S,39,C6X8,1,1,BLUE,;
COLOR:ACT/C,39,C6X8,1,1,WHITE,;
COLOR:ACT/S,40,C6X14,1,1,BLUE,;
COLOR:ACT/C,40,C6X14,1,1,WHITE,;
COLOR:ACT/S,41,C7X3,1,1,BLUE,;
COLOR:ACT/C,41,C7X3,1,1,WHITE,;
COLOR:ACT/S,42,C7X5,1,1,BLUE,;
COLOR:ACT/C,42,C7X5,1,1,WHITE,;
COLOR:ACT/S,43,C7X8,1,1,BLUE,;
COLOR:ACT/C,43,C7X8,1,1,WHITE,;
COLOR:ACT/S,44,C7X15,1,1,BLUE,;
COLOR:ACT/C,44,C7X15,1,1,WHITE,;
COLOR:ACT/S,45,C8X4,1,1,BLUE,;
COLOR:ACT/C,45,C8X4,1,1,WHITE,;
COLOR:ACT/S,46,C8X6,1,1,BLUE,;
COLOR:ACT/C,46,C8X6,1,1,WHITE,;
COLOR:ACT/S,47,C8X7,1,1,BLUE,;
COLOR:ACT/C,47,C8X7,1,1,WHITE,;
COLOR:ACT/S,48,C8X16,1,1,BLUE,;
COLOR:ACT/C,48,C8X16,1,1,WHITE,;
COLOR:ACT/S,49,C9X1,1,1,BLUE,;
COLOR:ACT/C,49,C9X1,1,1,WHITE,;
COLOR:ACT/S,50,C9X10,1,1,BLUE,;
COLOR:ACT/C,50,C9X10,1,1,WHITE,;
COLOR:ACT/S,51,C9X11,1,1,BLUE,;
COLOR:ACT/C,51,C9X11,1,1,WHITE,;
COLOR:ACT/S,52,C9X13,1,1,BLUE,;
COLOR:ACT/C,52,C9X13,1,1,WHITE,;
```

```
COLOR:ACT/S,53,C10X2,1,1,BLUE,;
COLOR:ACT/C,53,C10X2,1,1,WHITE,;
COLOR:ACT/S,54,C10X9,1,1,BLUE,;
COLOR:ACT/C,54,C10X9,1,1,WHITE,;
COLOR:ACT/S,55,C10X12,1,1,BLUE,;
COLOR:ACT/C,55,C10X12,1,1,WHITE,;
COLOR:ACT/S,56,C10X14,1,1,BLUE,;
COLOR:ACT/C,56,C10X14,1,1,WHITE,;
COLOR:ACT/S,57,C11X3,1,1,BLUE,;
COLOR:ACT/C,57,C11X3,1,1,WHITE,;
COLOR:ACT/S,58,C11X9,1,1,BLUE,;
COLOR:ACT/C,58,C11X9,1,1,WHITE,;
COLOR:ACT/S,59,C11X12,1,1,BLUE,;
COLOR:ACT/C,59,C11X12,1,1,WHITE,;
COLOR:ACT/S,60,C11X15,1,1,BLUE,;
COLOR:ACT/C,60,C11X15,1,1,WHITE,;
COLOR:ACT/S,61,C12X4,1,1,BLUE,;
COLOR:ACT/C,61,C12X4,1,1,WHITE,;
COLOR:ACT/S,62,C12X10,1,1,BLUE,;
COLOR:ACT/C,62,C12X10,1,1,WHITE,;
COLOR:ACT/S,63,C12X11,1,1,BLUE,;
COLOR:ACT/C,63,C12X11,1,1,WHITE,;
COLOR:ACT/S,64,C12X16,1,1,BLUE,;
COLOR:ACT/C,64,C12X16,1,1,WHITE,;
COLOR:ACT/S,65,C13X5,1,1,BLUE,;
COLOR:ACT/C,65,C13X5,1,1,WHITE,;
COLOR:ACT/S,66,C13X9,1,1,BLUE,;
COLOR:ACT/C,66,C13X9,1,1,WHITE,;
```

```
COLOR:ACT/S,67,C13X14,1,1,BLUE,;
COLOR:ACT/C,67,C13X14,1,1,WHITE,;
COLOR:ACT/S,68,C13X15,1,1,BLUE,;
COLOR:ACT/C,68,C13X15,1,1,WHITE,;
COLOR:ACT/S,69,C14X6,1,1,BLUE,;
COLOR:ACT/C,69,C14X6,1,1,WHITE,;
COLOR:ACT/S,70,C14X10,1,1,BLUE,;
COLOR:ACT/C,70,C14X10,1,1,WHITE,;
COLOR:ACT/S,71,C14X13,1,1,BLUE,;
COLOR:ACT/C,71,C14X13,1,1,WHITE,;
COLOR:ACT/S,72,C14X16,1,1,BLUE,;
COLOR:ACT/C,72,C14X16,1,1,WHITE,;
COLOR:ACT/S,73,C15X7,1,1,BLUE,;
COLOR:ACT/C,73,C15X7,1,1,WHITE,;
COLOR:ACT/S,74,C15X11,1,1,BLUE,;
COLOR:ACT/C,74,C15X11,1,1,WHITE,;
COLOR:ACT/S,75,C15X13,1,1,BLUE,;
COLOR:ACT/C,75,C15X13,1,1,WHITE,;
COLOR:ACT/S,76,C15X16,1,1,BLUE,;
COLOR:ACT/C,76,C15X16,1,1,WHITE,;
COLOR:ACT/S,77,C16X8,1,1,BLUE,;
COLOR:ACT/C,77,C16X8,1,1,WHITE,;
COLOR:ACT/S,78,C16X12,1,1,BLUE,;
COLOR:ACT/C,78,C16X12,1,1,WHITE,;
COLOR:ACT/S,79,C16X14,1,1,BLUE,;
COLOR:ACT/C,79,C16X14,1,1,WHITE,;
COLOR:ACT/S,80,C16X15,1,1,BLUE,;
COLOR:ACT/C,80,C16X15,1,1,WHITE,;
```

Bibliography

1. Wiley, P. "A Parallel Architecture Comes of Age at Last," *IEEE Spectrum*, 46-50 (June 1987).
2. Pritsker, A. *Introduction to Simulation and SLAM II*. New York: John Wiley, 1986.
3. Nestle, E. and Inselberg, A. "The Synapse N+1 System: Architectural Characteristics and Performance Data of a Tightly-Coupled Multiprocessor System," *Proceedings of the 12th Annual IEEE Computer Architecture Symposium*. 233-239. Silver Spring, Md: IEEE Press, June 1985.
4. Du, H. "On the Performance of Synchronous Multiprocessors," *IEEE Transactions on Computing, 34 (5)*: 462-466 (May 1985).
5. Bhuyan, L.N. "On the Performance of Loosely-Coupled Multiprocessors," *Proceedings of the 11th Annual IEEE Computer Architecture Symposium*. 256-262. New York: IEEE Press, June 1984.
6. Padmanabahn, K. and Lawrie, D. "Performance Analysis of Redundant-Path Networks for Multiprocessor Systems," *ACM Transactions on Computing Systems, 3 (2)*: 117-144 (May 1985).
7. Jalby, W. and Meier, U. "Optimizing Matrix Operations on a Parallel Multiprocessor with a Hierarchical Memory System," *Proceedings of the 1986 International Conference on Parallel Processing*. 429-432. New YorK: IEEE Press, 1986.
8. Vrsalovic, D. and others. "The Influence of Parallel Decomposition Strategies on the Performance of Multiprocessor Systems," *Proceedings of the 12th Annual IEEE Computer Architecture Symposium*. 396-405. Silver Spring, Md: IEEE Press, June 1985.
9. Vrsalovic, D. and others. "Performance Prediction for Multiprocessor Systems," *Proceedings of the 13th International Conference on Parallel Processing*. 139-146. New York: IEEE Press, Aug 1984.
10. Reed, D. and Grunwald, D. "The Performance of Multicomputer Interconnection Networks," *Computer*, 63-73 (June 1987).
11. SAS User's Guide: Statistics, Version 5 Edition. Cary, NC: SAS Institute Inc, 1985.
12. Standridge, Charles R. and A. Alan B. Pritsker, *TESS: The Extended Simulation System*. New York: Halstead Press, 1987.

Vita

Captain Timothy Scott Moore was born on June 4, 1961 in San Angelo, Texas. He graduated from high school in Southside, Alabama in 1979 and attended the University of Alabama, from which he received the degree of Bachelor of Science in Computer Science in May 1983. Upon graduation, he received a commission in the USAF through the ROTC program. He reported for active duty in October 1983 to Offutt AFB, Nebraska where he served as a Warning Systems Analyst and Command and Control Missile Warning Software Engineer for the Command Center Processing and Display System. He entered the School of Engineering, Air Force Institute of Technology, in June 1986. Captain Moore is a member of the Tau Beta Pi engineering honor society.

Permanent address: Route 1 Box 560
Gadsden, Alabama 35901

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A189 5-73

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GCS/ENG/87D-20	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION School of Engineering 6b OFFICE SYMBOL (If applicable) AFIT/ENG	7a NAME OF MONITORING ORGANIZATION
6c ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology (AU) Wright-Patterson AFB, Ohio 45433-6583	7b ADDRESS (City, State, and ZIP Code)
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

SIMULATION STUDY OF A PARALLEL PROCESSOR WITH UNBALANCED LOADS

12 PERSONAL AUTHOR(S)

Timothy S. Moore, B.S., Captain, USAF

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
MS Thesis	FROM ______ TO ______	1987 December	115

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Parallel Processing, Parallel Processor, Simulation
12	05		

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: A SIMULATION STUDY OF A PARALLEL PROCESSOR WITH UNBALANCED LOADS

Thesis Chairman: Wade H. Shaw, Captain, USA
Associate Professor of Electrical Engineering and
Computer Engineering

AFR 190-1
31 Dec 87
Development
Air Force Institute of Technology (AU)
Wright-Patterson AFB OH 45433

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL Wade H. Shaw, Captain, USA	22b TELEPHONE (Include Area Code) 513-255-3576 22c OFFICE SYMBOL AFIT/ENG

DD Form 1473, JUN 86 Previous editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

The purpose of this study was twofold; first, to estimate the impact of unbalanced computational loads on a parallel processing architecture via Monte Carlo simulation and second, to investigate the impact of representing the dynamics of the parallel processing problem via animated simulation. The study is constrained to the hypercube architecture. Two independent variables, the degree of imbalance and the degree of locality are defined. The degree of imbalance characterizes the nature or severity of the load imbalance and the degree of locality characterizes the node loadings with respect to node locations across the cube.

A SLAM II simulation model of a generic 16 node hypercube was constructed in which each node processes a predetermined number of computational tasks, and following each task, sends a message to a single randomly chosen receiver node. An experiment was designed in which the independent variables, degree of imbalance and degree of locality were varied across two computation-to-IO ratios to determine their separate and interactive affects on the dependent variable, job speedup.

ANOVA and regression techniques were used to estimate the relationship between load imbalance, locality, the computation-to-IO ratio, and their interactions to job speedup. The results show that load imbalance severely impacts a parallel processor's performance. The effect of locality is minor and enters the speedup model primarily as an interactive term; suggesting that the locality effect on speedup is dependent on the degree of imbalance. The intensity of IO is significant and affects speedup across all levels of locality and imbalance.

An animated simulation was developed using The Extended Simulation System (TESS) and the SLAM II model mentioned previously. The animation was designed such that a 16 node hypercube structure was displayed. The processing nodes and channels were displayed in different colors to represent specific types of processing. Watching the animation execute proved useful in two ways. First, the animation was useful in visually explaining the concepts of imbalance and locality. Secondly, and most importantly, the animation was valuable as a means of verifying the underlying simulation model.

END
DATE
3-88
DTIC